AD-A 089 985

AD

CONTRACT REPORT ARBRL-CR-00433

FINITE ELEMENT ANALYSIS OF A DYNAMICALLY LOADED FLAT LAMINATED PLATE

Prepared by

University of Illinois
Aeronautical and Astronautical Eng Dept
Urbana, Il 61801

July 1980

US ARMY ARMAMENT RESEARCH AND DEVELOPMENT COMMAND
BALLISTIC RESEARCH LABORATORY
ABERDEEN PROVING GROUND, MARYLAND

Approved for public release; distribution unlimited.

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
CONTRACT REPORT ARBRL-CR-00433

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
FINITE ELEMENT ANALYSIS OF A DYNAMICALLY LOADED FLAT LAMINATED PLATE

5. TYPE OF REPORT & PERIOD COVERED
Final Contractor Report
November 1976 to March 1979

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Adam R. Zak
Daniel W. Pillasch

8. CONTRACT OR GRANT NUMBER(s)
DAAD05-77-C-0722

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Aeronautical and Astronautical Engineering Dept.
University of Illinois, Urbana, Illinois 61801

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
USA Armament Research and Development Command
USA Ballistic Research Laboratory(ATTN:DRDAR-BL)
Aberdeen Proving Ground, MD 21005

12. REPORT DATE
July 1980

13. NUMBER OF PAGES

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Anisotropic
Analysis
Finite-element and
Three-dimensional deformation
Dynamic response
Plates Laminates

20. ABSTRACT (Continue on reverse side if necessary and identify by block number) (mba)

A finite element structural model has been developed for the dynamic analysis of laminated, thick plates. The model uses quadrilateral elements to represent the shape of the plate and the elements are stacked in the thickness direction to represent various material layers. This analysis allows for orthotropic, elastic-plastic or elastic-viscoplastic material properties. Non-linear strain displacement relations are used to represent large transverse plate deflections. A finite difference technique is used to perform the numerical time integration.

(continued)

(Item 20 continued)

Assumptions are made about the response through the thickness of the plate that allow for the use of relatively large time increments in the numerical integration method. These led to a set of element equilibrium equations which help to keep small error from affecting the values of stress that are calculated. The results from the analytical model are compared with experimental results. A good agreement is found between the calculated and measured values of the transverse plate deflections.

TABLE OF CONTENTS

I. INTRODUCTION

The objective of this analysis has been to develop a finite element model to analyze dynamically loaded, laminated, orthotropic thick plates with efficient numerical methods of solution. It is intended to apply the analysis to relatively large magnitude forces which are rapidly changing with time, consequently the materials are assumed to be represented by elastic-plastic and elastic-viscoplastic models. The finite element model assumes large transverse deflections of the plate which lead to nonlinear strain-displacement relations. Because the analysis is intended for predicting inter-laminar stresses and thickness changes, basic plate theory will not suffice. However, full three-dimensional analysis is much too cumbersome to use, so this method makes use of the plate-like characteristics, but still allows for three-dimensional response. The finite element model leads to a set of dynamic matrix equations representing the nodal plate displacements. The basic approach to the solution of these equations consists of an incremental approach in the time domain.[1]

The method of attack is to first calculate the internal forces from the incremental stresses and deflections of the previous time increment using the stiffness matrix. The external load is input and, using the theory of virtual work, it is transformed to external nodal forces. Using Newmark's beta finite difference technique,[2] the deflections for the new time increment are calculated. From these new deflections the stresses are calculated and these stresses are checked to see if elastic-plastic or elastic-viscoplastic yielding has occurred. The time is then incremented up and this technique marches on for the desired length of time. Figure 1 shows a flow diagram of the computer program with appropriate section references.

This report will first review the basic finite element formulation of the problem which will include the development of the stiffness matrix and the stress calculations. It will then discuss the development of the external nodal forces, the dynamic analysis with a variable time increment, and the orthotropic, elastic-plastic and elastic-viscoplastic analysis. The remainder of the analysis will be devoted to the methods used to increase the size of the time increment in the time integration and the element equilibrium equations that resulted from this increase in time increment size. Finally, a numerical example will be discussed, and the accuracy of this method of analysis will be compared to actual experimental results.

Review of Finite Element Formulation for Laminated Plates

Because of the physical characteristics of a thick laminated plate, it is natural to use Cartesian coordinates and to align the plane of the plate with the x_1 - x_2 coordinate plane and the thickness with the x_3 coordinate axis. The shape of the finite element will then be a general quadrilateral defined in the x_1 - x_2 coordinate plane with a constant thickness in the x_3 coordinate axis. To insure a homogeneous element, there should be as many elements stacked in the x_3 direction as there are different laminated layers. A typical numbering system and axis location is shown in Figure 2. The mass of each element is lumped at the nodes. The actual calculation of the mass at each node is calculated by bisecting each side of the triangular element and joining these center points to the center node which defines the apex of the four triangular elements in Figure 3. Thus, the original quadrilateral has been divided into four smaller ones which contain two of the eight nodes of the large element. The mass of each smaller quadrilateral portion of the element is found by multiplying the thickness of the element by the density by the area of the smaller quadrilateral element, and the result is distributed equally to the two nodes. The mass matrix is obtained by summing masses at each structural node from the adjacent elements. The resulting matrix is diagonal.

For ease in calculating the stiffness matrix, each quadrilateral element is subdivided into four triangular elements with the center node not carrying any load so that it can later be removed by static condensation when the triangular elements are summed to form the quadrilateral element. A typical element and sub-element nodal numbering system is shown in Figure 3.

Displacement Functions

The first step in finite element formulation is to choose a displacement function. Since each traingular element has three degrees of freedom at each node, the assumed displacement function for each element should have eighteen generalized coordinates. The displacement function, chosen to be linear in the x_1 - x_2 plane and to vary linearly in the x_3 direction, is shown below:

$$\begin{aligned}
u_1 &= \alpha_1 + \alpha_2 x_1 + \alpha_3 x_2 + x_3\ (\beta_1+\beta_2 x_1+\beta_3 x_2) \\
u_2 &= \alpha_4 + \alpha_5 x_1 + \alpha_6 x_2 + x_3\ (\beta_4+\beta_5 x_1+\beta_6 x_2) \\
u_3 &= \alpha_7 + \alpha_8 x_1 + \alpha_9 x_2 + x_3\ (\beta_7+\beta_8 x_1+\beta_9 x_2)
\end{aligned} \qquad 1.1$$

where u_i (i=1,3) are the Cartesian displacement components in terms of x_i directions and α_j and β_j (j=1,9), the eighteen unknown generalized coordinates. Writing Equations 1.1 at each of the six nodes of a triangular element results in eighteen equations which can be written as the matrix shown below:

$$
\begin{Bmatrix} {}^1u_1 \\ {}^1u_2 \\ {}^1u_3 \\ {}^2u_1 \\ {}^2u_2 \\ {}^2u_3 \\ {}^3u_1 \\ {}^3u_2 \\ {}^3u_3 \\ {}^4u_1 \\ {}^4u_2 \\ {}^4u_3 \\ {}^5u_1 \\ {}^5u_2 \\ {}^5u_3 \\ {}^6u_1 \\ {}^6u_2 \\ {}^6u_3 \end{Bmatrix}
=
\begin{bmatrix}
1 & {}^1x_1 & {}^1x_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & {}^1x_1 & {}^1x_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & {}^1x_1 & {}^1x_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & {}^2x_1 & {}^2x_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & {}^2x_1 & {}^2x_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & {}^2x_1 & {}^2x_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & {}^3x_1 & {}^3x_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & {}^3x_1 & {}^3x_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & {}^3x_1 & {}^3x_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & {}^4x_1 & {}^4x_2 & 0 & 0 & 0 & 0 & 0 & 0 & t & t\,{}^4x_1 & t\,{}^4x_2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & {}^4x_1 & {}^4x_2 & 0 & 0 & 0 & 0 & 0 & 0 & t & t\,{}^4x_1 & t\,{}^4x_2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & {}^4x_1 & {}^4x_2 & 0 & 0 & 0 & 0 & 0 & 0 & t & t\,{}^4x_1 & t\,{}^4x_2 \\
1 & {}^5x_1 & {}^5x_2 & 0 & 0 & 0 & 0 & 0 & 0 & t & t\,{}^5x_1 & t\,{}^5x_2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & {}^5x_1 & {}^5x_2 & 0 & 0 & 0 & 0 & 0 & 0 & t & t\,{}^5x_1 & t\,{}^5x_2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & {}^5x_1 & {}^5x_2 & 0 & 0 & 0 & 0 & 0 & 0 & t & t\,{}^5x_1 & t\,{}^5x_2 \\
1 & {}^6x_1 & {}^6x_2 & 0 & 0 & 0 & 0 & 0 & 0 & t & t\,{}^6x_1 & t\,{}^6x_2 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & {}^6x_1 & {}^6x_2 & 0 & 0 & 0 & 0 & 0 & 0 & t & t\,{}^6x_1 & t\,{}^6x_2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & {}^6x_1 & {}^6x_2 & 0 & 0 & 0 & 0 & 0 & 0 & t & t\,{}^6x_1 & t\,{}^6x_2
\end{bmatrix}
\begin{Bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \\ \alpha_7 \\ \alpha_8 \\ \alpha_9 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \\ \beta_6 \\ \beta_7 \\ \beta_8 \\ \beta_9 \end{Bmatrix}
\qquad 1.2
$$

where the forward superscripts on u_i and x_i quantities refer to one of the six nodes and t represents the element thickness. Equation 1.2 can be written in a short form as:

$$\{\delta\} = [E]\{\alpha\} \qquad 1.3$$

Solving for the global coordinates gives:

$$\{\alpha\} = [E]^{-1}\{\delta\} \qquad 1.4$$

Because there are so many zeros, matric [E] can be inverted by hand in the following manner. First, divide each column matrix in half so there is an upper and lower generalized coordinate matrix (α,β) and an upper and lower nodal displacement matrix (δ_u,δ_ℓ). This changes Equation 1.3 to:

$$\begin{Bmatrix} \{\delta_u\} \\ \{\delta_\ell\} \end{Bmatrix} = \begin{bmatrix} [E_1] & [0] \\ [E_2] & t[E_2] \end{bmatrix} \begin{Bmatrix} \{\alpha\} \\ \{\beta\} \end{Bmatrix} \qquad 1.5$$

where the [E] matrix has been divided into four parts. Inverting Equation 1.5 by parts yields:

$$\{\alpha\} = [E_1]^{-1}\{\delta_u\} \qquad 1.6$$

$$\{\beta\} = \frac{1}{t}[E_2]^{-1}\Big\{\{\delta_\ell\} - [E_2]\{\alpha\}\Big\} \qquad 1.7$$

Inserting Equation 1.6 into 1.7 yields:

$$\{\beta\} = \frac{1}{t}[E_2]^{-1}\{\delta_\ell\} - \frac{1}{t}[E_2]^{-1}[E_2][E_1]^{-1}\{\delta_u\} \qquad 1.8$$

Using Equations 1.6 and 1.8, it is apparent that Equation 1.4 can be written as:

$$\begin{Bmatrix} \{\alpha\} \\ \{\beta\} \end{Bmatrix} = \begin{bmatrix} [E_1]^{-1} & [0] \\ -\frac{1}{t}[E_1]^{-1} & \frac{1}{t}[E_2]^{-1} \end{bmatrix} \begin{Bmatrix} \{\delta_u\} \\ \{\delta_\ell\} \end{Bmatrix} \qquad 1.9$$

Using the following notation:

$$a_i = {}^{j}x_1\,{}^{m}x_2 - {}^{m}x_1\,{}^{j}x_2$$

$$b_i = {}^{j}x_2 - {}^{m}x_2$$

$$c_i = {}^{m}x_1 - {}^{j}x_1 \qquad 1.10$$

$$2A_{t \text{ or } b} = \det \begin{vmatrix} 1 & {}^{i}x_1 & {}^{i}x_2 \\ 1 & {}^{j}x_1 & {}^{j}x_2 \\ 1 & {}^{m}x_1 & {}^{m}x_2 \end{vmatrix}$$

where i,j,m are cyclic permutations either of 1 to 3 for b (bottom of the element) or 4 to 6 for t (top of the element), the individual matrices are as follows:

$$[E_1]^{-1} = \frac{1}{2A_b} \begin{bmatrix} a_1 & 0 & 0 & a_2 & 0 & 0 & a_3 & 0 & 0 \\ b_1 & 0 & 0 & b_2 & 0 & 0 & b_3 & 0 & 0 \\ c_1 & 0 & 0 & c_2 & 0 & 0 & c_3 & 0 & 0 \\ 0 & a_1 & 0 & 0 & a_2 & 0 & 0 & a_3 & 0 \\ 0 & b_1 & 0 & 0 & b_2 & 0 & 0 & b_3 & 0 \\ 0 & c_1 & 0 & 0 & c_2 & 0 & 0 & c_3 & 0 \\ 0 & 0 & a_1 & 0 & 0 & a_2 & 0 & 0 & a_3 \\ 0 & 0 & b_1 & 0 & 0 & b_2 & 0 & 0 & b_3 \\ 0 & 0 & c_1 & 0 & 0 & c_2 & 0 & 0 & c_3 \end{bmatrix} \qquad 1.11$$

$$[E_2]^{-1} = \frac{1}{2A_t} \begin{bmatrix} a_4 & 0 & 0 & a_5 & 0 & 0 & a_6 & 0 & 0 \\ b_4 & 0 & 0 & b_5 & 0 & 0 & b_6 & 0 & 0 \\ c_4 & 0 & 0 & c_5 & 0 & 0 & c_6 & 0 & 0 \\ 0 & a_4 & 0 & 0 & a_5 & 0 & 0 & a_6 & 0 \\ 0 & b_4 & 0 & 0 & b_5 & 0 & 0 & b_6 & 0 \\ 0 & c_4 & 0 & 0 & c_5 & 0 & 0 & c_6 & 0 \\ 0 & 0 & a_4 & 0 & 0 & a_5 & 0 & 0 & a_6 \\ 0 & 0 & b_4 & 0 & 0 & b_5 & 0 & 0 & b_6 \\ 0 & 0 & c_4 & 0 & 0 & c_5 & 0 & 0 & c_6 \end{bmatrix} \qquad 1.12$$

Strain-Displacement Relation

The next step in the finite element formulation is to calculate the internal work due to a virtual change in nodal displacement. The strains must first be found in terms of the generalized coordinates. The non-linear strain displacement relations in tensor notation are:

$$\varepsilon_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i} + u_{k,i}u_{k,j}) \qquad 1.13$$

Making use of the fact that this analysis is for a plate, it is assumed that deflections and rotations out of the plane of the plate are large compared to those in the plane of the plate, or symbolically:

$$\begin{aligned} &u_3 > u_1, u_2 \\ &u_{3,1} > u_{1,1}, u_{2,1} \\ &u_{3,2} > u_{1,2}, u_{2,2} \\ &u_{1,3}, u_{2,3} > u_{3,3} \end{aligned} \qquad 1.14$$

Using the order of magnitude defined in Equations 1.14 reduces Equations 1.13 to the following form:

$$\begin{aligned} \varepsilon_{11} &= u_{1,1} + \frac{1}{2} u^2_{3,1} \\ \varepsilon_{22} &= u_{2,2} + \frac{1}{2} u^2_{3,2} \\ \varepsilon_{33} &= u_{3,3} + \frac{1}{2} (u^2_{1,3} + u^2_{2,3}) \\ \varepsilon_{12} &= \frac{1}{2} (u_{1,2} + u_{2,1} + u_{3,1}u_{3,2}) \\ \varepsilon_{23} &= \frac{1}{2} (u_{2,3} + u_{3,2}) \\ \varepsilon_{13} &= \frac{1}{2} (u_{3,1} + u_{1,3}) \end{aligned} \qquad 1.15$$

Since there are linear and nonlinear terms, Equation 1.15 can be re-written in matrix notation as:

$$\{\varepsilon\} = \{\varepsilon_{\ell}\} + \{\varepsilon_{n\ell}\} \tag{1.16}$$

where:

$$\{\varepsilon_{\ell}\} = \frac{1}{2}(u_{i,j} + u_{j,i}) \tag{1.17}$$

and:

$$\{\varepsilon_{n\ell}\} = \frac{1}{2}[u_{3,1}^2, u_{3,2}^2, u_{1,3}^2 + u_{2,3}^2, u_{3,1}u_{3,2}, 0, 0]^T \tag{1.18}$$

Performing the indicated operations in Equation 1.17 on 1.1 results in:

$$\begin{Bmatrix} \varepsilon_{11\ell} \\ \varepsilon_{22\ell} \\ \varepsilon_{33\ell} \\ \varepsilon_{12\ell} \\ \varepsilon_{23\ell} \\ \varepsilon_{13\ell} \end{Bmatrix} = \begin{bmatrix} 0&1&0&0&0&0&0&0&0&0&x_3&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0&0&0&x_3&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&x_1&x_2 \\ 0&1&0&0&0&0&0&0&0&0&x_3&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0&0&0&x_3&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&x_1&x_2 \end{bmatrix} \{\alpha\} \tag{1.19}$$

In matrix notation Equation 1.19 reduces to:

$$\{\varepsilon_{\ell}\} = [Q]\,\{\alpha\} \tag{1.20}$$

Substituting Equation 1.4 into 1.20 results in:

$$\{\varepsilon_{\ell}\} = [Q]\,[E]^{-1}\{\delta\} \tag{1.21}$$

Taking virtual changes, this can be written as:

$$d\,\{\varepsilon_{\ell}\} = [B_{\ell}]\,d\,\{\delta\} \tag{1.22}$$

where:

$$[B_{\ell}] = [Q]\,[E]^{-1} \tag{1.23}$$

and the symbol d in front of a matrix represents a virtual change.

Disregarding the zero terms, Equation 1.18 can be written as:

$$\begin{Bmatrix} \varepsilon_{11n\ell} \\ \varepsilon_{22n\ell} \\ \varepsilon_{33n\ell} \\ \varepsilon_{12n\ell} \end{Bmatrix} = \frac{1}{2} \begin{bmatrix} u_{3,1} & 0 & 0 & 0 \\ 0 & u_{3,2} & 0 & 0 \\ 0 & 0 & u_{1,3} & u_{2,3} \\ u_{3,2} & u_{3,1} & 0 & 0 \end{bmatrix} \begin{Bmatrix} u_{3,1} \\ u_{3,2} \\ u_{1,3} \\ u_{2,3} \end{Bmatrix} \qquad 1.24$$

In matrix notation Equation 1.24 reduces to:

$$\{\varepsilon_{n\ell}\} = \frac{1}{2} [A] \{\Theta\} \qquad 1.25$$

Since the virtual changes are considered in the finite element method, it is necessary to investigate virtual changes in the nonlinear strain which can be written as follows:

$$d \{\varepsilon_{n\ell}\} = \frac{1}{2} [A] d \{\Theta\} + \frac{1}{2} d [A] \{\Theta\} \qquad 1.26$$

Clearly from Equation 1.24:

$$[A] d \{\Theta\} = d [A] \{\Theta\} \qquad 1.27$$

From Equations 1.26 and 1.27, the following result is true:

$$d \{\varepsilon_{n\ell}\} = [A] d \{\Theta\} \qquad 1.28$$

Performing the indicated operations in Equation 1.24 on 1.1 results in:

$$d \{\Theta\} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_3 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_3 \\ 0 & 0 & 1 & x_1 & x_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & x_1 & x_2 & 0 & 0 \end{bmatrix} d \begin{Bmatrix} \alpha_8 \\ \alpha_9 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \\ \beta_6 \\ \beta_8 \\ \beta_9 \end{Bmatrix} \qquad 1.29$$

and:

$$[A] = \begin{bmatrix} \alpha_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta_8 & 0 \\ 0 & \alpha_9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta_9 \\ 0 & 0 & \beta_1 & \beta_2 & \beta_3 & \beta_4 & \beta_5 & \beta_6 & 0 & 0 \\ \alpha_9 & \alpha_8 & 0 & 0 & 0 & 0 & 0 & 0 & \beta_9 & \beta_8 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & x_1 & 0 \\ 0 & 0 & x_2 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & x_1 \\ 0 & 0 & 0 & x_2 \\ x_3 & 0 & 0 & 0 \\ 0 & x_3 & 0 & 0 \end{bmatrix} \quad 1.30$$

Combining Equations 1.28, 1.29 and 1.30 results in:

$$d\{\varepsilon_{n\ell}\} = \begin{bmatrix} \alpha_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta_8 & 0 \\ 0 & \alpha_9 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \beta_9 \\ 0 & 0 & \beta_1 & \beta_2 & \beta_3 & \beta_4 & \beta_5 & \beta_6 & 0 & 0 \\ \alpha_9 & \alpha_8 & 0 & 0 & 0 & 0 & 0 & 0 & \beta_9 & \beta_8 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_3 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_3 \\ 0 & 0 & 1 & x_1 & x_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & x_1 & x_1^2 & x_1x_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & x_2 & x_1x_2 & x_2^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & x_1 & x_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & x_1 & x_2^2 & x_1x_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & x_2 & x_1x_2 & x_2^2 & 0 & 0 \\ x_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_3^2 & 0 \\ 0 & x_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_3^2 \end{bmatrix} d\begin{Bmatrix} \alpha_8 \\ \alpha_9 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \\ \beta_6 \\ \beta_8 \\ \beta_9 \end{Bmatrix} \qquad 1.31$$

Equation 1.31 can be written in a short form as follows:

$$d\,\{\varepsilon_{n\ell}\} = [\bar{\alpha}]\,[Z]\,d\,\{\bar{\alpha}\} \qquad 1.32$$

where the matrices $[\bar{\alpha}]$, $[Z]$ and $d\{\bar{\alpha}\}$ are symbolic representation of the three matrices on the right hand side of Equation 1.31. Since $[\bar{\alpha}]$ are constants which can be written in terms of the rows of $[E]^{-1}$ corresponding to the correct α_j and $\{\delta\}$ given in Equation 1.4, Equation 1.32 can be rewritten as:

$$d\,\{\varepsilon_{n\ell}\} = [\bar{\alpha}]\,[Z]\,[\bar{E}]\,d\,\{\delta\} \qquad 1.33$$

where $[\bar{E}]$ also consists of the rows corresponding to the correct α_j in $\{\bar{\alpha}\}$. This is written as the normal strain-displacement relationship below:

$$d\,\{\varepsilon_{n\ell}\} = [B_{n\ell}]\,d\,\{\delta\} \qquad 1.34$$

where:

$$[B_{n\ell}] = [\bar{\alpha}]\,[Z]\,[\bar{E}] \qquad 1.35$$

Note that $[B_n\ ]$ is dependent on displacements from [ ]. From Equations 1.16, 1.22 and 1.34, the complete strain-displacement relationship becomes:

$$d\,\{\varepsilon\} = [B_{\ell}]\,d\,\{\delta\} + [B_{n\ell}]\,d\,\{\delta\} \qquad 1.36$$

which in general is written as:

$$d\,\{\varepsilon\} = [B]\,d\,\{\delta\} \qquad 1.37$$

where:

$$[B] = [B_{\ell}] + [B_{n\ell}] \qquad 1.38$$

Stress-Strain Relationship

The only other calculation to be made before the virtual work is evaluated is the stress-strain relationship. Since this analysis deals with orthotropic material, only the orthotropic relationship will be discussed since in the limit these equations become isotropic when the modulus of elasticity and Poisson's ratio are input into the following equations:

$$\varepsilon_{11}' = \frac{1}{E_1}[\sigma_{11}'-\nu_{12}\sigma_{22}'-\nu_{13}\sigma_{33}']$$

$$\varepsilon_{22}' = \frac{1}{E_2}[\sigma_{22}'-\nu_{21}\sigma_{11}'-\nu_{23}\sigma_{33}']$$

$$\varepsilon_{33}' = \frac{1}{E_3}[\sigma_{33}'-\nu_{31}\sigma_{11}'-\nu_{32}\sigma_{22}'] \qquad 1.39$$

$$\varepsilon_{12}' = \frac{\sigma_{12}'}{2G_{12}}$$

$$\varepsilon_{23}' = \frac{\sigma_{23}'}{2G_{23}}$$

$$\varepsilon_{13}' = \frac{\sigma_{13}'}{2G_{13}}$$

where E_i is the modulus of elasticity of the x_i direction, ν_{ij} and G_{ij} are the Poisson's ratio and shear modulus along the x_i-x_j plane. It should be noted that the prime in these equations refers to the local numbering system. Due to the plane of symmetry for orthotropic materials, the following is true:

$$\frac{\nu_{12}}{E_1} = \frac{\nu_{21}}{E_2}$$

$$\frac{\nu_{13}}{E_1} = \frac{\nu_{31}}{E_3} \qquad 1.40$$

$$\frac{\nu_{23}}{E_2} = \frac{\nu_{32}}{E_3}$$

Letting:

$$\zeta = 1-\nu_{32}\nu_{23}-\nu_{12}\nu_{21}-\nu_{12}\nu_{23}\nu_{31}-\nu_{13}\nu_{32}\nu_{21} \qquad 1.41$$

the inverse of Equation 1.39 is:

$$\sigma_{11}' = \frac{1}{\zeta}[(1-\nu_{32}\nu_{23})E_1\varepsilon_{11}'+(\nu_{12}+\nu_{13}\nu_{23})E_2\varepsilon_{22}'+(\nu_{13}+\nu_{12}\nu_{23})E_3\varepsilon_{33}']$$

$$\sigma_{22}' = \frac{1}{\zeta}[(\nu_{21}+\nu_{23}\nu_{31})E_1\varepsilon_{11}'+(1-\nu_{13}\nu_{31})E_2\varepsilon_{22}'+(\nu_{23}+\nu_{21}\nu_{13})E_3\varepsilon_{33}']$$

$$\sigma_{33}' = \frac{1}{\zeta}[(\nu_{31}+\nu_{32}\nu_{21})E_1\varepsilon_{11}'+(\nu_{32}+\nu_{31}\nu_{12})E_2\varepsilon_{22}'+(1-\nu_{12}\nu_{21})E_3\varepsilon_{33}']$$

$$\sigma_{12}' = 2G_{12}\varepsilon_{12}' \qquad 1.42$$

$$\sigma_{23}' = 2G_{23}\varepsilon_{23}'$$

$$\sigma_{13}' = 2G_{13}\varepsilon_{13}'$$

In matrix notation Equation 1.42 is:

$$\{\sigma'\} = [C'] \{\varepsilon'\} \qquad 1.43$$

Often, though, the axes for which the material properties are defined do not align with those of the global coordinate system. When this happens, the stresses of interest are those of the global system. Letting the global system have the unprimed stresses and strains, an equation similar to Equation 1.43 can be written as follows:

$$\{\sigma\} = [C] \{\varepsilon\} \qquad 1.44$$

In order to transform $\{\sigma'\}$ to $\{\sigma\}$ the direction cosines between x_i and x_j' can be used. We define directional cosines by α_{ij} as follows:

$$\alpha_{ij} = \cos (x_i, x_j') \qquad 1.45$$

and therefore from stress transformation relations, it follows that:

$$\sigma_{k\ell}' = \alpha_{ik}\alpha_{j\ell}\sigma_{ij} \qquad 1.46$$

and:

$$\varepsilon_{k\ell}' = \alpha_{ik}\alpha_{j\ell}\varepsilon_{ij} \qquad 1.47$$

Equations 1.46 and 1.47 can be rewritten in matrix form as:

$$\{\sigma'\} = [R]\{\sigma\} \qquad 1.48$$

and:

$$\{\varepsilon'\} = [R]\{\varepsilon\} \qquad 1.49$$

where:

$$[R] = \begin{bmatrix} \alpha_{11}^2 & \alpha_{21}^2 & \alpha_{31}^2 & 2\alpha_{11}\alpha_{21} & 2\alpha_{31}\alpha_{21} & 2\alpha_{11}\alpha_{31} \\ \alpha_{12}^2 & \alpha_{22}^2 & \alpha_{32}^2 & 2\alpha_{12}\alpha_{22} & 2\alpha_{32}\alpha_{22} & 2\alpha_{12}\alpha_{32} \\ \alpha_{13}^2 & \alpha_{23}^2 & \alpha_{33}^2 & 2\alpha_{13}\alpha_{23} & 2\alpha_{33}\alpha_{23} & 2\alpha_{13}\alpha_{33} \\ \alpha_{11}\alpha_{12} & \alpha_{21}\alpha_{22} & \alpha_{31}\alpha_{32} & \alpha_{11}\alpha_{22}+\alpha_{21}\alpha_{12} & \alpha_{11}\alpha_{32}+\alpha_{31}\alpha_{12} & \alpha_{21}\alpha_{32}+\alpha_{31}\alpha_{22} \\ \alpha_{12}\alpha_{13} & \alpha_{22}\alpha_{23} & \alpha_{32}\alpha_{33} & \alpha_{12}\alpha_{23}+\alpha_{22}\alpha_{13} & \alpha_{22}\alpha_{33}+\alpha_{32}\alpha_{23} & \alpha_{12}\alpha_{33}+\alpha_{32}\alpha_{13} \\ \alpha_{11}\alpha_{13} & \alpha_{21}\alpha_{23} & \alpha_{31}\alpha_{33} & \alpha_{11}\alpha_{23}+\alpha_{21}\alpha_{13} & \alpha_{21}\alpha_{33}+\alpha_{31}\alpha_{23} & \alpha_{11}\alpha_{33}+\alpha_{31}\alpha_{13} \end{bmatrix} \qquad 1.50$$

The work done in either coordinate system must be equal, therefore:

$$d\,\{\varepsilon\}^T\{\sigma\} = d\,\{\varepsilon'\}^T\{\sigma'\} \qquad 1.51$$

Substituting Equations 1.43, 1.44 and 1.49 into 1.51 results in:

$$d\,\{\varepsilon\}^T [C]\{\varepsilon\} = \{\varepsilon\}^T [R]^T [C'] [R]\{\varepsilon\} \qquad 1.52$$

Therefore, from Equation 1.52 the stress-strain matrix in global coordinates can be written in terms of the material stress-strain relationship and the transformation matrices. This can be written as:

$$[C] = [R]^T [C'] [R] \qquad 1.53$$

In the computer program the transformation matrix is found by two separate transformations. Figure 4 shows the orientation of the coordinate systems and the two angles needed to describe the transformation.

Note that x_2' lies in the x_1-x_2 plane. Using the α and β described here, not to be confused with the subscripted generalized coordinates, the transformation matrices can be calculated by the appropriate insertion into the direction cosines.

Internal Force

It is now possible to calculate the virtual change in the internal work of the structure due to a virtual change in the nodal displacements. This can be written as:

$$d\,W_I = \int_V d\,\{\varepsilon\}^T\,\{\sigma\}\,dV \qquad 1.54$$

Using Equations 1.37 in 1.54, the internal work is written as:

$$d\,W_I = d\,\{\delta\}^T \int_V [B]^T\,\{\sigma\}\,dV \qquad 1.55$$

In addition to the work done by internal work of the finite element, there is external work done by the forces at the eight corners. Denoting these forces by the matrix {f}, the external virtual work is:

$$d\,W_E = d\,\{\delta\}^T\,\{f\} \qquad 1.56$$

Since external and internal forces on each element are in equilibrium, it follows from virtual work that:

$$d\,W_I + d\,W_E = 0 \qquad 1.57$$

Therefore, from Equations 1.55, 1.56 and 1.57:

$$\{f\} = -\int_V [B]^T\,\{\sigma\}\,dV \qquad 1.58$$

The forces acting on any node resulting from the adjacent elements are obtained from Equation 1.58. This summation will result in a matrix of internal forces denoted by $\{F_I\}$.

The next step in the analysis is to obtain the nodal equilibrium equations for the total structure. Because this is a dynamic problem the inertial effects must be considered. The equation of motion is:

$$[M]\{\ddot{\Delta}\} = \{F_I\} + \{F_E\} \quad 1.59$$

where [M] is the diagonal mass matrix previously described, $\{\ddot{\Delta}\}$ is the global displacement acceleration, and $\{F_E\}$ is the concentrated nodal force matrix to be described in Section 2. The solution of Equation 1.59 will be discussed in Section 3.

II. EXTERNAL FORCE

Because of the varying types of distributed loads that could be used in a problem of this type, a technique was developed that only needed the input of the distributed external load at the nodal locations that the load acts on. This is normally accomplished by adding a subroutine to the program which gives that data. Using the theory of virtual work it is possible to transform these external distributed loads to the effective concentrated forces $\{F_E\}$ which are used in Equation 1.59.

Since the top of the plate which carries the distributed loads is composed of triangular elements, as shown in Figure 3, the simplest representation of the distributed load L over the plate is:

$$L = \xi_1 + \xi_2 x_1 + \xi_3 x_2 \qquad 2.1$$

where ξ_i are some suitable parameters defining the linear distribution. Writing Equation 2.1 at the three nodes results in:

$$\begin{Bmatrix} L_1 \\ L_2 \\ L_3 \end{Bmatrix} = \begin{bmatrix} 1 & {}^1x_1 & {}^1x_2 \\ 1 & {}^2x_1 & {}^2x_2 \\ 1 & {}^3x_1 & {}^3x_2 \end{bmatrix} \begin{Bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{Bmatrix} \qquad 2.2$$

where forward superscripts in x_i refer to nodal locations. Using the notation defined by Equation 1.10, the inverse of Equation 2.1 is written in the following form:

$$\begin{Bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{Bmatrix} = \frac{1}{2A} \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix} \begin{Bmatrix} L_1 \\ L_2 \\ L_3 \end{Bmatrix} \qquad 2.3$$

Substituting Equation 2.3 into 2.1 yields:

$$L = \frac{1}{2A}[1x_1x_2]\begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \\ c_1 & c_2 & c_3 \end{bmatrix}\begin{Bmatrix} L_1 \\ L_2 \\ L_3 \end{Bmatrix} \qquad 2.4$$

Letting:

$$\begin{aligned} [a] &= [a_1a_2a_3] \\ [b] &= [b_1b_2b_3] \\ [c] &= [c_1c_2c_3] \\ \{L_i\} &= \begin{Bmatrix} L_1 \\ L_2 \\ L_3 \end{Bmatrix} \end{aligned} \qquad 2.5$$

Substituting 2.5 into 2.4 results in:

$$L = \frac{1}{2A}([a] + [b]x_1 + [c]x_2)\{L_i\} \qquad 2.6$$

Referring to Equation 1.1 it is apparent that when x_3 is a constant,

$$u_3 = \xi_1 + \xi_2x_1 + \xi_3x_2 \qquad 2,7$$

Following a similar procedure to that above results in:

$$u_3 = \frac{1}{2A}([a] + [b]x_1 + [c]x_2)\{u_{3_i}\} \qquad 2.8$$

where $\{u_{3_i}\}$ are nodal displacements of node i in the x_3 direction. The virtual work of the distributed load is:

$$dW_L = \int_A du_3^T \, L \, d \, A \qquad 2.9$$

Substituting Equations 2.6 and 2.7 into 2,9 results in:

$$dW_L = \int_A d\,\{u_{3_i}\}^T\frac{1}{2A}([a]+[b]x_1+[c]x_2)^T\frac{1}{2A}([a]+[b]x_1+[c]x_2)\{L_i\}dA \qquad 2.10$$

The virtual work done by the distributed load L is now replaced by effective nodal forces which do virtual work as given by:

$$dW_E = d\,\{u_{3_i}\}^T \{F_E\} \tag{2.11}$$

where W_E represents the work done by the effective concentrated forces F_E. Since the effective forces are caused by the distributed loads, it follows that:

$$dW_E = dW_L \tag{2.12}$$

Substituting Equations 2.10 and 2.11 into 2.12 gives:

$$\{F_E\} = \frac{1}{4A^2} [D] \{L_i\} \tag{2.13}$$

where:

$$[D] = [a]^T [D_1] + [b]^T [D_2] + [c]^T [D_3]$$

$$[D_1] = [a] + [b] \int_A x_1 dA + [c] \int_A x_2 dA$$

$$[D_2] = [a] \int_A x_1 dA + [b] \int_A x_2^2 \, dA + [c] \int_A x_1 x_2 dA \tag{2.14}$$

$$[D_3] = [a] \int_A x_2 dA + [b] \int_A x_1 x_2 dA + [c] \int_A x_2^2 dA$$

The reason for this method of writing the final result is that this method was found computationally more efficient to calculate the final forces. Section 6 will discuss the loading used in the test examples. Knowing how to calculate the internal and external forces, the next step is the solution of the nodal equilibrium equations.

III. DYNAMIC ANALYSIS

Because the use of a variable time increment would be advantageous in analyzing a structure with a load applied over a specific length of time, a method of analysis was developed to include the variable time increment. This gives the programmer the responsibility to choose appropriate time increments over the length of time that the structure is analyzed. Good engineering judgment will dictate the size of these time increments. When there is a rapid change in external load, the time increments should be small when compared to times when there is less change in load or no load at all. Normally the initial analysis should be very accurate so even smaller time increments could be used. The numerical solution of Equation 1.59 by the finite difference method involves the replacement of the time derivatives by their finite difference equivalents.[2,3] For convenience the matrix notation will be dropped and the displacement symbol (Δ) will be replaced by the quantity x with a subscript defining the time interval. In this approach the time history is divided into discrete time intervals whose length will be denoted by h with a subscript defining the time interval. From the general theory of kinematics, the velocity and displacement relations are written for the n^{th} and $(n+1)^{th}$ time interval as follows:

$$\dot{x}_n = \dot{x}_{n-1} + \frac{h_n}{2}(\ddot{x}_{n-1} + \ddot{x}_n) \qquad 3.1$$

$$x_n = x_{n-1} + h_n\dot{x}_{n-1} + (\frac{1}{2} - \beta)h_n^2\ddot{x}_{n-1} + \beta h_n^2\ddot{x}_n \qquad 3.2$$

$$x_{n+1} = x_n + h_{n+1}\dot{x}_n + (\frac{1}{2} - \beta)h_{n+1}^2\ddot{x}_n + \beta h_{n+1}^2\ddot{x}_{n+1} \qquad 3.3$$

where $\ddot{x}$, $\dot{x}$ and x represent acceleration, velocity, and displacement at time intervals denoted by the subscript.

Equation 3.1 means that the velocity at the end of the interval is equal to the sum of the velocity at the beginning of the interval and the product of the time interval and the average of the acceleration at the beginning and end of the interval. Equation 3.2 and similarly Equation 3.3 are obtained by integrating Equation 3.1 and introducing a weighted acceleration parameter, β, to express the average acceleration.

For example, $\beta=1/4$ gives a linear estimation for the average acceleration. The coefficient β is chosen so as to best represent the system being analyzed. Newmark[2] discusses convergence and the values of beta and shows that for dynamic problems, $\beta=1/4$ gives an infinite stability limit and is the value used for most dynamic problems. At

$t = \sum_{k=1}^{n+1} h_k, \sum_{k=1}^{n} h_k, \sum_{k=1}^{n-1} h_k$ respectively, the equations of motion become:

$$M\,\ddot{x}_{n+1} = F_{n+1} \qquad 3.4$$

$$M\,\ddot{x}_{n} = F_{n} \qquad 3.5$$

$$M\,\ddot{x}_{n-1} = F_{n-1} \qquad 3.6$$

$$F = F_I + F_E \qquad 3.7$$

The unknowns in Equations 3.1 through 3.6 are x_{n+1}, $\dot{x}_n$, $\dot{x}_{n-1}$, $\ddot{x}_{n+1}$, $\ddot{x}_n$, $\ddot{x}_{n-1}$. Equations 3.3 will be used to solve for displacement at time n+1. Equations 3.4, 3.5, 3.6 give values of the accelerations in terms of known forces and masses. The only unknown then is $\dot{x}_n$. Rearranging Equation 3.1 gives:

$$\dot{x}_n = \frac{h_n}{h_n}\dot{x}_{n-1} + \frac{h_n^2}{2h_n}\ddot{x}_{n-1} + \frac{h_n}{2}\ddot{x}_n + \beta h_n\ddot{x}_{n-1} - \beta\frac{h_n^2}{h_n}\ddot{x}_{n-1} + \beta\frac{h_n^2}{h_n}\ddot{x}_n - \beta h_n\ddot{x}_n \qquad 3.8$$

This can be rewritten as:

$$\dot{x}_n = \frac{1}{h_n}[h_n\dot{x}_{n-1} + (\tfrac{1}{2}-\beta)h_n^2\ddot{x}_{n-1} + \beta h_n^2\ddot{x}_n] + \beta h_n\ddot{x}_{n-1} + (\tfrac{1}{2}-\beta)h_n\ddot{x}_n \qquad 3.9$$

Combining Equations 3.2 and 3.9 results in the following:

$$\dot{x}_n = \frac{1}{h_n}(x_n - x_{n-1}) + \beta h_n\ddot{x}_{n-1} + (\tfrac{1}{2}-\beta)h_n\ddot{x}_n \qquad 3.10$$

Inserting Equation 3.10 into 3.3 yields:

$$x_{n+1}=x_n+\frac{h_{n+1}}{h_n}(x_n-x_{n-1})+(h_{n+1}+h_n)h_{n+1}(\frac{1}{2}-\beta)\ddot{x}_n+\beta h^2_{n+1}\ddot{x}_{n+1}+\beta h_{n+1}h_n\ddot{x}_{n-1} \tag{3.11}$$

Substituting Equations 3.4, 3.5, 3.6, 3.7 into 3.11 gives:

$$x_{n+1} = x_n+\frac{h_{n+1}}{h_n}(x_n-x_{n-1})-(h_{n+1}+h_n)h_{n+1}(\frac{1}{2}-\beta)M^{-1}(F_{I_n}+F_{E_n})$$

$$+\beta h^2_{n+1}M^{-1}(F_{I_{n+1}}+F_{E_{n+1}})+\beta h_n h_{n+1}M^{-1}(F_{I_{n-1}}+F_{E_{n-1}}) \tag{3.12}$$

Using the original matrix notation this becomes:

$$\{\Delta\}_{n+1}=\{\Delta\}_n+\frac{h_{n+1}}{h_n}(\{\Delta\}_n-\{\Delta\}_{n-1})+h_{n+1}(h_{n+1}+h_n)(\frac{1}{2}-\beta)[M]^{-1}(\{F_I\}_n+\{F_E\}_n)$$

$$+\beta h^2_{n+1}[M]^{-1}(\{F_I\}_{n+1}+\{F_E\}_{n+1})+\beta h_{n+1}h_n[M]^{-1}(\{F_I\}_{n-1}+\{F_E\}_{n-1}) \tag{3.13}$$

For this method to work, it is assumed that the internal forces are known. But to know the internal forces, the stresses must be known as indicated in Equation 1.58. To know incremental stress, the strain must be known. To know strains, deflections must be known. If the time steps are small enough, then there is not much change in stress from one time increment to another. Thus, this method is extremely accurate. But for larger time increments another approach must be used. This is described in Sections 4 and 6.

IV. MODIFICATION FOR LARGE TIME INCREMENTS

The limitations on the size of the integration time step are a direct result of the finite element model which has a lumped mass at each node. As the numerical integration proceeds, the masses move relative to each other. If the time step is too large, the relative motion of the masses is exaggerated and artificial oscillation is induced. In order to combat this artificial oscillation, the x_1-x_2 deflections are coupled and the deflections in the x_3 direction are coupled through the thickness.

First, then, the in-plane (u_1-u_2) displacements will be discussed. In order to have a more convenient equation to work with, Equation 3.13 is written with a constant time increment as follows:

$$\{\Delta\}_{n+1} = 2\{\Delta\}_n-\{\Delta\}_{n-1}+\beta h^2[M]^{-1}[\{F_I\}_{n+1}+(\frac{1}{\beta}-2)\{F_I\}_n+\{F_I\}_{n-1}]$$

$$+\beta h^2[M]^{-1}[\{F_E\}_{n+1}+(\frac{1}{\beta}-2)\{F_E\}_n+\{F_E\}_{n-1}] \qquad 4.1$$

where $\{\Delta\}$ is the displacement matrix, β is the acceleration parameter, h is the time interval, $\{F_I\}$ is the internal force matrix, $\{F_E\}$ is the external force matrix, [M] is the mass matrix, and the subscripts n, n-1, and n+1 denote time intervals.

In analyzing the u_1 and u_2 displacements, it is assumed that they are linearly dependent through the thickness. This forces plane sections to remain plane.

This first assumption results in the following equations:

$$u_1 = q_1 + zq_2$$

$$u_2 = q_3 + zq_4 \qquad 4.2$$

where q_k, k=1, 4, are unknown generalized coefficients called the transformed displacements and z is the distance in the x_3 direction of the node from the center of gravity. The importance of having z be the distance from the center of gravity will be discussed when the transformed mass matrix is discussed. In matrix notation Equation 4.2 becomes:

$$\{\Delta\} = [tf]\{q\} \qquad 4.3$$

where {q} is the matrix of transformed displacements and [tf] is the transformation matrix described below. Letting ℓ be the number of layers of material and $i=\ell$, $1+\ell$ be the nodal location in the thickness direction, the transformation matrix can be written as:

$$[tf] = \begin{vmatrix} [tf_1] \\ [tf_2] \\ \cdot \\ [tf_i] \\ \cdot \\ [tf_{\ell+1}] \end{vmatrix} \qquad 4.4$$

where:

$$[tf_i] = \begin{vmatrix} 1 & z_i & 0 & 0 \\ 0 & 0 & 1 & z_i \end{vmatrix} \qquad 4.5$$

and z_i is the distance of node i from the center of gravity. The notation i=1 is the node at the bottom of the plate and $i=1+\ell$ is the node at the top.

Since the displacements are written in terms of transformed displacements, the forces should be written in terms of the transformed forces. Letting $\{f_E\}$ be the matrix of external forces corresponding to {q}, and $\{f_I\}$ be the internal forces also corresponding to {q}, the principle of virtual work states:

$$d\ \{\Delta\}^T\ \{F_I\} = d\ \{q\}^T\ \{f_I\} \qquad 4.6$$

Transposing Equation 4.3 yields:

$$\{\Delta\}^T = \{q\}^T\ [tf]^T \qquad 4.7$$

Substituting Equation 4.7 into 4.6 yields:

$$d\ \{q\}^T\ [tf]^T\ \{F_I\} = d\ \{q\}^T\ \{f_I\} \qquad 4.8$$

Therefore:

$$\{f_I\} = [tf]^T \{F_I\} \quad 4.9$$

Similarly:

$$\{f_E\} = [tf]^T \{F_E\} \quad 4.10$$

Finding a transformed mass matrix [m] starts with:

$$-[M] \{\ddot{\Delta}\} = \{F\}_{inertia} \quad 4.11$$

By virtual work:

$$-d \{\Delta\}^T [M] \{\ddot{\Delta}\} = d\{\Delta\}^T \{F\}_{inertia} \quad 4.12$$

$$-d \{q\}^T [m] \{\ddot{q}\} = d\{\Delta\}^T \{F\}_{inertia} \quad 4.13$$

Therefore:

$$d \{\Delta\}^T [M] \{\ddot{\Delta}\} = d\{q\}^T [m] \{\ddot{q}\} \quad 4.14$$

Substituting Equations 4.3 and 4.7 into 4.14 yields:

$$d \{q\}^T [m] \{\ddot{q}\} = d\{q\}^T [tf]^T[M][tf]\{\ddot{q}\} \quad 4.15$$

Dividing out the unnecessary terms gives:

$$[m] = [tf]^T [M] [tf] \quad 4.16$$

Now the reason for z_i being the distance from the center of gravity will become apparent. The calculations are much simplified by the mass matrix being a diagonal matrix as in the case for the original mass matrix. The original mass matrix was:

$$[M] = [M_i] \quad 4.17$$

Performing the matrix multiplication in Equation 4.16 using Equations 4.4, 4.5 and 4.7 produces:

$$[m] = \begin{bmatrix} \sum_{i=\ell}^{1+\ell} M_i & \sum_{i=\ell}^{1+\ell} M_i z_i & 0 & 0 \\ \sum_{i=\ell}^{1+\ell} M_i z_i & \sum_{i=\ell}^{1+\ell} M_i z_i^2 & \sum_{i=\ell}^{1+\ell} M_i z_i & 0 \\ 0 & \sum_{i=\ell}^{1+\ell} M_i z_i & \sum_{i=\ell}^{1+\ell} M_i & \sum_{i=\ell}^{1+\ell} M_i z_i \\ 0 & 0 & \sum_{i=\ell}^{1+\ell} M_i z_i & \sum_{i=\ell}^{1+\ell} M_i z_i^2 \end{bmatrix} \qquad 4.18$$

But,

$$\sum_{i=\ell}^{1+\ell} M_i z_i = 0 \qquad 4.19$$

by the definition of the center of gravity, thus causing [m] to be a diagonal matrix.

Referring to Equation 4.1, the only elements that still must be transformed are the deflections for past time intervals. From Equation 4.3:

$$\{q\}_n = [tf]^{-1} \{\Delta\}_n \qquad 4.20$$

To find these generalized displacements, it is only necessary to know four of the actual displacements to solve for four unknowns. This is easily done by hand and results in:

$$\begin{bmatrix} [tf_1] \\ [tf_2] \end{bmatrix}^{-1} = \begin{bmatrix} -\frac{z_2}{z_1-z_2} & 0 & \frac{z_1}{z_1-z_2} & 0 \\ \frac{1}{z_1-z_2} & 0 & -\frac{1}{z_1-z_2} & 0 \\ 0 & -\frac{z_2}{z_1-z_2} & 0 & \frac{z_1}{z_1-z_2} \\ 0 & \frac{1}{z_1-z_2} & 0 & -\frac{1}{z_1-z_2} \end{bmatrix} \qquad 4.21$$

Substituting the transformed quantities into Equation 4.1 yields:

$$\{q\}_{n+1} = 2\{q\}_n - \{q\}_{n-1} + \beta h^2 [m]^{-1} \left| \{f_I\}_n + \left(\frac{1}{\beta} - 2\right) \{f_I\}_{n-1} + \{f_I\}_{n-2} \right|$$

$$+ \beta h^2 [m]^{-1} \left| \{f_E\}_{n+1} + \left(\frac{1}{\beta} - 2\right) \{f_E\}_n + \{f_E\}_{n-1} \right| \qquad 4.22$$

It should be noted that the internal transformed forces are displaced by one time increment. Because these forces are small, even at large time increments, this yields accurate results with small error. Equation 4.22 is thus solved and Equation 4.3 transforms the results to global displacements.

Although this time lag is acceptable for the in-plane displacements, it is not acceptable for the u_3 displacements. The reason for this is that the external force is being applied in the u_3 direction, thus making these internal forces more sensitive to larger time intervals. In order to account for the change in the internal force, a model was sought to couple the deflections through the thickness.

In finding a model to represent what happens through the thickness, it is necessary to see what the unknowns are. From Equation 4.1 the unknowns are $\{\Delta\}_{n+1}$ and $\{F_I\}_{n+1}$. All the other terms are known. In order to predict what $\{F_I\}_{n+1}$ is, it is necessary to couple the deflections through the thickness and to assume all strains small when compared to the strain in the u_3 direction. This can be done by letting:

$$\{F_I\}_{n+1} = \{F_I\}_n + \{\Delta F_I\}_{n+1} \qquad 4.23$$

where $\{\Delta F_I\}_{n+1}$ is the change of the internal force between time intervals. The deflection in the u_3 direction is then coupled by the model shown in Figure 5. This model assumes the stiffness between the nodal points in the thickness direction is much greater than the stiffness between in-plane nodal points. As long as the external force is in the u_3 direction this is a good assumption.

From Figure 5:

$$\Delta F_{I_{i,n+1}} = k_i [(\Delta_{i+1,n+1} - \Delta_{i,n+1}) - (\Delta_{i+1,n} - \Delta_{i,n})] - k_{i-1}$$

$$[(\Delta_{i,n+1} - \Delta_{i-1,n+1}) - (\Delta_{i,n} - \Delta_{i-1,n})] \qquad 4.24$$

where i refers to the nodal location through the thickness and n refers to the time increment. The predicted stiffness (k_i) is found from the orthotropic properties (C_{ij}, i=1,6, j=1,6). In matrix notation:

$$\begin{Bmatrix}\sigma_{11}\\ \sigma_{22}\\ \sigma_{33}\\ \sigma_{12}\\ \sigma_{23}\\ \sigma_{13}\end{Bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & 0 & 0 & 0\\ C_{21} & C_{22} & C_{23} & 0 & 0 & 0\\ C_{31} & C_{32} & C_{33} & 0 & 0 & 0\\ 0 & 0 & 0 & C_{44} & 0 & 0\\ 0 & 0 & 0 & 0 & C_{55} & 0\\ 0 & 0 & 0 & 0 & 0 & C_{66}\end{bmatrix} \begin{Bmatrix}\varepsilon_{11}\\ \varepsilon_{22}\\ \varepsilon_{33}\\ \varepsilon_{12}\\ \varepsilon_{23}\\ \varepsilon_{13}\end{Bmatrix} \qquad 4.24$$

Using the assumption that all strains are small when compared to the strain in the u_3 direction yields:

$$\sigma_{33} = C_{33}\varepsilon_{33} \qquad 4.26$$

This gives the stiffness for a unit cube equal to C_{33}. Therefore:

$$k_i = \frac{C_{33_i}\,\mathrm{Area}_i}{t_i} \qquad 4.27$$

where t_i is the thickness of layer i, Area_i is the area used to compute the mass of the nodal point, and C_{33_i} is the orthotropic property of the quadrilateral that the node lies in. Letting:

$$\Delta'_{i,n+1} = 2\Delta_{i,n} + \Delta_{i,n-1} + \frac{\beta h^2}{M_i}\left[F_{I_{i,n}} + \left(\frac{1}{\beta} - 2\right) F_{I_{i,n}} + F_{I_{i,n-1}}\right] + \frac{\beta h^2}{M_i}\left[F_{E_{i,n+1}} + \left(\frac{1}{\beta} - 2\right) F_{E_{i,n}} + F_{E_{i,n-1}}\right] \qquad 4.28$$

and substituting Equations 4.23 and 4.28 into 4.1 produces:

$$\Delta_{i,n+1} = \Delta^{*}_{i,n+1} + \frac{\beta h^2}{M_i} [k_i(\Delta_{i+1,n+1}-\Delta_{i,n+1}) - k_{i-1}(\Delta_{i,n+1}-\Delta_{i-1,n+1})$$

$$- k_i(\Delta_{i+1,n}-\Delta_{i,n}) + k_{i-1}(\Delta_{i,n}-\Delta_{i-1,n})] \qquad 4.29$$

where the only unknowns are the deflections at time n+1. This produces i=1+ℓ (ℓ is the number of layers) number of simultaneous equations which can be solved for. The technique of coupling the motion of the masses in the thickness direction, as described in this section, permits the use of larger time increments than otherwise possible. The effect of this coupling is to reduce the numerical oscillations produced by excessive, relative motion of the adjacent masses. However, the use of larger time increments introduces an additional side effect for rapidly varying external loads. If these loads vary by a great amount from one time interval to another, then the transverse stresses from the n^{th} time interval will be greatly out of balance with the external forces at $(n+1)^{th}$ interval. This leads to large force unbalance which can lead to numerical errors. The Section VI of the report will discuss a method for eliminating these large differences.

V. PLASTICITY

Orthotropic Elastic-Plastic Yielding

The present analysis allows for permanent deformation to occur in the structure. Because of the practical considerations each finite element is assumed to be either elastic or plastic. The general method of attack is to calculate the stresses as if they were elastic. Using Hill's orthotropic yield criterion,[4] the stresses are checked to see if they are physically compatible with the yield criterion. If they are compatible the analysis continues with those values of stress. If they are not compatible, then plastic flow has occurred and using the flow rule and the yield criterion, the stresses are recalculated to account for this plastic behavior.

The first step then is to calculate the stresses. Using tensor notation, the incremental form of the stress-strain equation for elastic case gives the stress change from time t_n to t_{n+1}:

$$d\sigma_{ij}^{T} = C_{ijkl}\, d\varepsilon_{kl} \qquad 5.1$$

where the superscript T indicates a test value. The total stress at a time (t_{n+1}):

$$\sigma_{ij}^{T} = \sigma_{ij_n} + d\sigma_{ij}^{T} \qquad 5.2$$

where the stress σ_{ij_n} is from time t_n.
This value is then put into Hill's yield criterion which requires six yield stresses (Y_{ij}). Before showing the yield criterion, the following constants are defined:

$$\overline{Y}_{11} = \frac{1}{Y_{11}^2} - \frac{1}{Y_{22}^2} - \frac{1}{Y_{33}^2}$$

$$\overline{Y}_{22} = \frac{1}{Y_{22}^2} - \frac{1}{Y_{11}^2} - \frac{1}{Y_{33}^2}$$

$$\overline{Y}_{33} = \frac{1}{Y_{33}^2} - \frac{1}{Y_{11}^2} - \frac{1}{Y_{22}^2} \qquad 5.3$$

Then the yield criterion can be written as:

$$2f(\sigma_{ij}) = \frac{\sigma_{11}^2}{Y_{11}^2} + \frac{\sigma_{22}^2}{Y_{22}^2} + \frac{\sigma_{33}^2}{Y_{33}^2} + \frac{\sigma_{12}^2}{Y_{12}^2} + \frac{\sigma_{23}^2}{Y_{23}^2} + \frac{\sigma_{13}^2}{Y_{13}^2} + \overline{Y}_{11}\sigma_{22}\sigma_{33} + \overline{Y}_{22}\sigma_{11}\sigma_{33}$$

$$+ \overline{Y}_{33}\sigma_{11}\sigma_{22} = 1 \qquad 5.4$$

Using Equation 5.2 in 5.4, and if $2f(\sigma_{ij}^T) \leq 1$, then σ_{ij}^T represents the real stress at t_{n+1}; but if $2f(\sigma_{ij}^T) > 1$, yield has occurred. To calculate the stress for plastic flow, the strain increment is divided into elastic (ε^e) and plastic (ε^p) strain.

$$d\varepsilon_{ij} = d\varepsilon_{ij}^e + d\varepsilon_{ij}^p \qquad 5.5$$

The flow rule is written in the following manner:

$$d\varepsilon_{11}^p = d\lambda \left(\frac{\sigma_{11}}{Y_{11}^2} + \frac{\overline{Y}_{22}\sigma_{33}+\overline{Y}_{33}\sigma_{22}}{2}\right) = d\lambda T_{11}$$

$$d\varepsilon_{22}^p = d\lambda \left(\frac{\sigma_{22}}{Y_{22}^2} + \frac{\overline{Y}_{11}\sigma_{33}+\overline{Y}_{33}\sigma_{11}}{2}\right) = d\lambda T_{22}$$

$$d\varepsilon_{33}^p = d\lambda \left(\frac{\sigma_{33}}{Y_{33}^2} + \frac{\overline{Y}_{11}\sigma_{22}+\overline{Y}_{22}\sigma_{11}}{2}\right) = d\lambda T_{33}$$

$$d\varepsilon_{12}^p = d\lambda \frac{\sigma_{12}}{Y_{12}^2} = d\lambda T_{12}$$

$$d\varepsilon^p_{23} = d\lambda \frac{\sigma_{23}}{Y^2_{23}} = d\lambda T_{23}$$

$$d\varepsilon^p_{13} = d\lambda \frac{\sigma_{13}}{Y^2_{13}} = d\lambda T_{13} \qquad 5.6$$

where T_{ij} are defined by Equations 5.6 and, for the purpose of numerical calculations, they are assumed to be given by the stresses from time t_n. In short:

$$d\varepsilon^p_{ij} = d\lambda T_{ij} \qquad 5.7$$

In view of Equation 5.5, the actual form of Equation 5.1 is:

$$d\sigma_{ij} = C_{ijkl} d\varepsilon^e_{kl} \qquad 5.8$$

Substituting Equation 5.5 into 5.6 results in:

$$d\sigma_{ij} = C_{ijkl}(d\varepsilon_{kl} - d\varepsilon^p_{kl}) \qquad 5.9$$

Now Equations 5.1 and 5.7 are substituted into 5.9 giving:

$$d\sigma_{ij} = d\sigma^T_{ij} - C_{ijkl} d\lambda T_{kl} \qquad 5.10$$

Using Equation 5.10 to update the stress at t_n gives stress at t_{n+1}:

$$\sigma_{ij} = \sigma^T_{ij} - d\lambda C_{ijkl} T_{kl} \qquad 5.11$$

Letting $\overline{T}_{ij} = C_{ijkl} T_{kl}$, Equation 5.11 becomes:

$$\sigma_{ij} = \sigma^T_{ij} - d\lambda \overline{T}_{ij} \qquad 5.12$$

Substituting Equation 5.12 into 5.4 results in a quadradic equation in $d\lambda$ as follows:

$$A d\lambda^2 - 2B d\lambda + C = 0 \qquad 5.13$$

where:

$$A = \frac{\overline{T}_{11}^2}{Y_{11}^2} + \frac{\overline{T}_{22}^2}{Y_{22}^2} + \frac{\overline{T}_{33}^2}{Y_{33}^2} + \frac{\overline{T}_{12}^2}{Y_{12}^2} + \frac{\overline{T}_{23}^2}{Y_{23}^2} + \frac{\overline{T}_{13}^2}{Y_{13}^2} + \overline{Y}_{11}\overline{T}_{22}\overline{T}_{33} + \overline{Y}_{22}\overline{T}_{11}\overline{T}_{33} + \overline{Y}_{33}\overline{T}_{22}\overline{T}_{11}$$

$$B = \frac{\overline{T}_{11}\sigma_{11}^T}{Y_{11}^2} + \frac{\overline{T}_{22}\sigma_{22}^T}{Y_{22}^2} + \frac{\overline{T}_{33}\sigma_{33}^T}{Y_{33}^2} + \frac{\overline{T}_{12}\sigma_{12}^T}{Y_{12}^2} + \frac{\overline{T}_{23}\sigma_{23}^T}{Y_{23}^2} + \frac{\overline{T}_{13}\sigma_{13}^T}{Y_{13}^2}$$

$$+ \overline{Y}_{11}\left(\frac{\overline{T}_{22}\sigma_{33}^T + \overline{T}_{33}\sigma_{22}^T}{2}\right) + \overline{Y}_{22}\left(\frac{\overline{T}_{11}\sigma_{33}^T + \overline{T}_{33}\sigma_{11}^T}{2}\right) + \overline{Y}_{33}\left(\frac{\overline{T}_{11}\sigma_{22}^T + \overline{T}_{22}\sigma_{11}^T}{2}\right)$$

$$C = 2f(\sigma_{ij}^T) - 1 \qquad 5.14$$

Solving Equation 5.13 results in:

$$d\lambda = \frac{2B \pm \sqrt{(2B)^2 - 4AC}}{2A} \qquad 5.15$$

Because C is defined in Equation 5.14 as the yield criterion, as C approaches 0, $d\lambda$ approaches 0. Clearly, the minus sign is the only sign that is physically acceptable. Multiplying top and bottom of Equation 5.15 by $B+\sqrt{B^2-AC}$ produces:

$$d\lambda = \frac{C}{B+\sqrt{B^2-AC}} \qquad 5.16$$

This $d\lambda$ is substituted into Equation 5.12 to give the actual stress at time t_{n+1}.

Orthotropic Elastic-Viscoplastic Yielding

Because of the nature of an impulsive load acting on a material, the plastic yielding is such that it is history dependent. To be able to handle materials with viscous coefficients, a viscoplastic analysis has been developed.

In this analysis it is assumed that the strain is divided into elastic ($d\varepsilon^{e}$) and viscoplastic ($d\varepsilon^{vp}$) strain.

$$d\varepsilon_{ij} = d\varepsilon_{ij}^{e} + d\varepsilon_{ij}^{vp} \qquad 5.17$$

Using Hill's Flow Rule, Equation 5.7 is modified to be:

$$d\varepsilon_{ij}^{vp} = d\lambda T_{ij} \qquad 5.18$$

It should be noted that the viscoplastic strain changes satisfy the incompressibility condition:

$$\sum_{i=1}^{3} d\varepsilon_{ii}^{vp} \qquad 5.19$$

The quantities T_{ij} represent six independent quantities and they can be arranged in a matrix form and then can be related to a stress matrix as follows:

$$\{T\} = [H]\{\sigma\} \qquad 5.20$$

where:

$$[H] = \begin{bmatrix} \frac{1}{Y_{11}^2} & \frac{\bar{Y}_{33}}{2} & \frac{\bar{Y}_{22}}{2} & 0 & 0 & 0 \\ \frac{\bar{Y}_{33}}{2} & \frac{1}{Y_{22}^2} & \frac{\bar{Y}_{11}}{2} & 0 & 0 & 0 \\ \frac{\bar{Y}_{22}}{2} & \frac{\bar{Y}_{11}}{2} & \frac{1}{Y_{33}^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{Y_{12}^2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{Y_{23}^2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{Y_{13}^2} \end{bmatrix} \qquad 5.21$$

By comparing the flow rule in Equation 5.18 to an isotropic case, it can be noted that the quantities T_{ij} have the same role as the deviatoric stresses and ε_{ij}^{vp} strains as the deviatoric strains. In fact, it may be noted that Equation 5.18 reduces in the limit to the isotropic case when the material properties are the same in all three directions. Consequently, following the procedure developed for the Bingham material,[6] the strain rate dependence is introduced by defining $\{T^F\}$:

$$\{T^F\} = \{T\} + \eta\{\dot{\varepsilon}^{vp}\} \tag{5.22}$$

where η represents viscous coefficient, and $\{T\}$ is the quantity which satisfies the yield criterion. By using Equation 5.20, a final stress is defined as:

$$\{\sigma^F\} = [H]^{-1}\{T^F\} \tag{5.23}$$

If no yielding has occurred, the viscoplastic strain increment is zero. So this analysis begins with a trial incremental stress where:

$$\{d\sigma^T\} = [C]\{d\varepsilon\} \tag{5.24}$$

and [C] is the orthotropic relationship between stress and strain. Equation 5.24 is inserted into Equation 5.2 and this value is inserted into the yield criterion in Equation 5.4. If yield does not occur, the trial stress is equal to σ^F. However, if yield does occur then Equations 5.22 and 5.23 must be used to calculate σ^F as follows:

From Equation 5.17 and 5.24:

$$\{d\sigma^F\} = [C]\{d\varepsilon\}-[C]\{d\varepsilon^{vp}\} \tag{5.25}$$

Then as in Equation 5.2:

$$\{\sigma^F\} = \{\sigma^T\} - [C]\{d\varepsilon^{vp}\} \tag{5.26}$$

Multiplying Equation 5.22 by $[H]^{-1}$ and using Equation 5.23 and 5.24, one gets:

$$\{\sigma^F\} = \{\sigma\} + \eta[H]^{-1}\{\dot{\varepsilon}^{vp}\} \tag{5.27}$$

From Equation 5.26 and 5.27, when solving for $\{\sigma\}$ one finds:

$$\{\sigma\} = \{\sigma^T\} - [C]\{d\varepsilon^{vp}\}-\eta[H]^{-1}\{\dot{\varepsilon}^{vp}\} \tag{5.28}$$

By using Equation 5.28 it is easily shown that:

$$\{\dot{\varepsilon}^{vp}\} = \{\frac{d\varepsilon^{vp}}{dt}\} = \frac{1}{\Delta t}\{d\varepsilon^{vp}\} = \frac{d\lambda}{\Delta t}\{T\} \qquad 5.29$$

Substituting 5.18 and 5.29 into 5.28 produces:

$$\{\sigma\} = \{\sigma^T\} - d\lambda[C]\{T\} - \frac{\eta d\lambda}{\Delta t}[H]^{-1}\{T\} \qquad 5.30$$

Defining another variable $\{\overline{T}\}$:

$$\{\overline{T}\} = [C]\ \{T\} \qquad 5.31$$

and using the inverse of Equation 5.20 and Equations 5.31 in 5.30 yields:

$$\{\sigma\} = \{\sigma^T\} - d\lambda \left|\{\overline{T}\} + \frac{\eta}{\Delta t}\{\sigma^T\}\right| \qquad 5.32$$

The question in a dynamic problem always arises as to what value of stress is used for the flow rule. In this formulation the stress used in the flow rule is approximated by the trial stress. A closer approximation can be formed by doing an iterative loop on this equation, but little difference is found in the solution when this is done.

The $\{\sigma\}$ stress formed here in Equation 5.32 is input in the yield criterion and results in the following equation:

$$Ad\lambda^2 - 2Bd\lambda + C = 0 \qquad 5.33$$

where:

$$T^{\cdot}_{ij} = \overline{T}^T_{ij} + \frac{\eta}{\Delta t}\sigma^T \qquad 5.34$$

$$A = \frac{T^{\cdot 2}_{11}}{Y^2_{11}} + \frac{T^{\cdot 2}_{22}}{Y^2_{22}} + \frac{T^{\cdot 2}_{33}}{Y^2_{33}} + \frac{T^{\cdot 2}_{12}}{Y^2_{12}} + \frac{T^{\cdot 2}_{23}}{Y^2_{23}} + \frac{T^{\cdot 2}_{13}}{Y^2_{13}} + \overline{Y}_{11}T^{\cdot}_{22}T^{\cdot}_{33} + \overline{Y}_{22}T^{\cdot}_{11}T^{\cdot}_{33} + \overline{Y}_{33}T^{\cdot}_{22}T^{\cdot}_{11} \qquad 5.35$$

$$B = \frac{T^{\cdot}_{11}\sigma^T_{11}}{Y^2_{11}} + \frac{T^{\cdot}_{22}\sigma^T_{22}}{Y^2_{22}} + \frac{T^{\cdot}_{33}\sigma^T_{33}}{Y^2_{33}} + \frac{T^{\cdot}_{12}\sigma^T_{12}}{Y^2_{12}} + \frac{T^{\cdot}_{23}\sigma^T_{23}}{Y^2_{23}} + \frac{T^{\cdot}_{13}\sigma^T_{13}}{Y^2_{13}}$$

$$+\overline{Y}_{11}\left(\frac{\dot{T}_{22}\sigma^T_{33}+\dot{T}_{33}\sigma^T_{22}}{2}\right)+\overline{Y}_{22}\left(\frac{\dot{T}_{11}\sigma^T_{33}+\dot{T}_{33}\sigma^T_{11}}{2}\right)+\overline{Y}_{33}\left(\frac{\dot{T}_{11}\sigma^T_{22}+\dot{T}_{22}\sigma^T_{11}}{2}\right) \qquad 5.36$$

$$C = 2\ f\ (\sigma^T_{ij}) - 1 \qquad 5.37$$

where $2f(\sigma^2_{ij})=1$ is the yield function.

The proportionality constant $d\lambda$ can be solved for and, as shown for Equation 5.13, is:

$$d\lambda = \frac{C}{B+\sqrt{B^2-AC}} \qquad 5.38$$

This value of $d\lambda$ is then substituted into Equation 5.32. By using Equations 5.28, 5.20 and 5.27:

$$\{\sigma^F\} = \left|1 + \frac{\eta d\lambda}{\Delta t}\right|\{\sigma\} \qquad 5.39$$

With this analysis completed, the problem of the oscillating stress is reduced but still sometimes occurs. This then leads to the next section which deals with the element equilibrium equations.

VI. ELEMENT EQUILIBRIUM EQUATIONS

The previous analysis helped allow for the use of larger time increments in the numerical time integration. Although the results showed the deflection to be stable, there were normal stress oscillations. In order to compensate for these small errors in strain, the external pressure is used to develop element equilibrium equations.

The modification of the analysis involves the introduction of additional dynamic equations which express the transverse acceleration of the elements as a whole, in addition to the nodal accelerations as expressed by Equation 1.56. Consider a small quadrilateral section of the plate as illustrated in Figure 6. In general, this section will be composed of N layers and loaded by external distributed force $p(x_1,x_2,t)$. The transverse acceleration of this plate section that is accelerating in the x_3 direction, is produced by two distinct forces. The first of these is the external force $p(x_1,x_2,t)$, and the second is the bending force in the plate. As a matter of observation, it is logical to expect the external force to be predominant in the early stages of the time history while the bending forces should become more important as the deformation of the plate increases. Since the amount of each contribution will vary with time, the relative amounts of acceleration from each load will become one of the unknowns of the problem. In order to take this into consideration, it is assumed that the amount of acceleration due to the external force will be related to the total average acceleration by a relation:

$$a_p = k\, a_t \tag{6.1}$$

where a_p is the acceleration due to the force $p(x_1,x_2,t)$, a_t is the total acceleration, and k is an unknown parameter. It may be noted that the accelerations in Equation 6.1 will vary through the plate thickness. Consider now the acceleration in the x_3 direction of the individual layers in the plate sections of Figure 6. These layers are now the actual finite elements. The normal stress in the x_3 direction will vary from layer to layer, and the top and bottom values of this stress will be equal to the applied surface loads. The plate elements through the thickness of the plate are illustrated in Figure 7. The interface stresses are denoted by $\overline{\sigma}$ with an appropriate subscript. Writing the dynamic equations for each layer results in the following N equations:

$$\bar{\sigma}_2 - \bar{\sigma}_I = a_{p_1} \rho_I t_I$$

$$\bar{\sigma}_3 - \bar{\sigma}_2 = a_{p_2} \rho_2 t_2$$

$$\text{-------------------------} \qquad 6.2$$

$$\bar{\sigma}_N - \bar{\sigma}_{N-1} = a_{p_{N-1}} \rho_{N-1} t_{N-1}$$

$$\bar{\sigma}_{N+1} - \bar{\sigma}_N = a_{p_N} \rho_N t_N$$

where N represents the total number of layers,   is the material density and t is the thickness of each layer. It may be noted from the boundary values that:

$$\bar{\sigma}_1 = 0 \qquad 6.3$$

$$\bar{\sigma}_{N+1} = p(x_1, x_2 t)$$

Using Equations 6.1 and 6.3 and adding Equation 6.2 results in the solution of the unknown constant k:

$$k = \frac{p(x_1, x_2, t)}{\sum_{j=1}^{N} a_{t_j} \rho_j t_j} \qquad 6.4$$

Since the finite element is formulated in terms of element stresses rather than the interface stresses, the latter have to be expressed in terms of the element stresses. This is done by linear extrapolation:

$$\sigma_j = \frac{(\bar{\sigma}_j + \bar{\sigma}_{j-1})}{2} \qquad 6.5$$

where σ_j represents the stress σ_{33} (normal stress in the x_3 direction). By using Equations 6.4 and 6.5 in Equation 6.2 it is possible to solve for the element stresses σ_1 to σ_N. This procedure will yield the normal stress in the x_3 direction for each element in the plate.

In applying this analysis to the existing finite element model, it must be realized that Equations 6.2 are not completely new dynamic equations; but rather the information contained in these equations is already contained, in a different form, in the original dynamic equation represented by Equation 1.59. However, Equations 1.59 represent nodal accelerations, but the accelerations in Equations 6.2 are some average accelerations in the elements and therefore the two sets of accelerations are related but are different quantities. In fact, in the present analysis the element accelerations are defined by averaging the nodal values. In order to see how Equations 6.2 are incorporated into the analysis, it is useful to review how the original analysis proceeded. Equations 1.59 are solved by a finite difference procedure and the resulting displacements are used to evaluate strains and therefore stresses. The stresses are then used to check for yield and to evaluate the body forces for the next time interval. In the modified approach Equations 1.59 still solve for the displacements which are used to evaluate all the strains. The displacements from Equations 1.59 are also used to evaluate the element accelerations which are needed for solution of Equations 6.2. The element accelerations are related to the nodal displacements by averaging over all the corner nodes for each element.

The value of the normal stress as calculated by this method is stored, and the previous values of stress are used to solve for the plastic yielding. After the completion of the plastic yielding, the stresses are modified so that they remain on the yield surface, yet comply with the element equilibrium equations. This is easily accomplished by referring to Hill's yield criterion (Equation 5.4) and rewriting it as:

$$2f(\sigma_{ij}) = -\frac{\overline{Y}_{11}}{2}(\sigma_{22}-\sigma_{33})^2 - \frac{\overline{Y}_{22}}{2}(\sigma_{33}-\sigma_{11})^2 - \frac{\overline{Y}_{33}}{2}(\sigma_{11}-\sigma_{22})^2$$

$$+\frac{\sigma_{12}^2}{Y_{12}^2} + \frac{\sigma_{23}^2}{Y_{23}^2} + \frac{\sigma_{13}^2}{Y_{23}^2} = 1 \qquad 6.6$$

If the superscript N refers to values found by nodal equilibrium equations and E refers to the value found by the elements equilibrium equations, the final stresses are solved for by the following equations:

$$\sigma_{11} = \sigma^N_{11} - \sigma^N_{33} + \sigma^E_{33}$$

$$\sigma_{22} = \sigma^N_{22} - \sigma^N_{33} + \sigma^E_{33}$$

$$\sigma_{33} = \sigma^E_{33} \tag{6.7}$$

$$\sigma_{12} = \sigma^N_{12}$$

$$\sigma_{23} = \sigma^N_{23}$$

$$\sigma_{13} = \sigma^N_{13}$$

This analysis was then programmed and the results compared favorably with an experiment. These findings and conclusions are discussed in the next section.

VII. RESULTS AND CONCLUSIONS

In order to check the accuracy of the previous analysis, runs were made with this program that inputted the requirements of the following experiment. The example used to check the program's accuracy was a three-layered laminated plate.[5] The top and bottom layers were 1020 steel, 12.7 mm and 6.35 mm thick, respectively. The middle layer was 2024 aluminum, 12.7 mm thick. The plate was 22.86 cm by 45.72 cm. The ends were simply supported and a large impulsive load was detonated on the top of the plate. The results of the experiment compared favorably to another computer code (acronym HEMP). The HEMP code is a Lagrangian finite difference technique that utilised an elastic-perfectly plastic model for solids. It is relatively time consuming and costly to run, thus the need for a more efficient technique such as the one previously described.

The input to this program which analyzed three-dimensional impulsively loaded plates (acronym TIP) is shown in Appendix A and is followed by the actual input used. As was mentioned in Section 2, in order to find the external force, a special subroutine must be input which gives the external pressure at specific locations. For this example a list of data was given for the external pressure at radial distances from the center of the plate at discrete time intervals. Figure 8 shows the data graphically for several times. Rather than input this data and extrapolate answers, the pressure was averaged over the plate at discrete time intervals using the formula:

$$p_{ave} = \sum_{i=1}^{N} \frac{p_i (r_i^2 - r_{i-1}^2)}{r_N^2} \qquad 7.1$$

where p_i is the pressure at radial distance r_i.

This gives an area type average and Figure 9 shows graphically how the results look. p_{ave} was then input in SUBROUTINE DISFOR; and knowing the time, the proper pressure is extrapolated from these average values. This cuts down considerably on the necessary input into SUBROUTINE DISFOR.

With the DISFOR and the input shown at the end of Appendix B, the program was executed with an elastic-plastic yield subroutine and executed again with an elastic-viscoplastic subroutine.

In the program with the elastic-plastic subroutine, the results shown in Figure 10 were achieved when using a time step of two and one half microseconds for the first twenty microseconds, then five microseconds for twenty to sixty microseconds, and ten microseconds for the remaining time. There was a four by six grid on the plate which is quite simple. The results shown here have a maximum error of five percent, which in all cases was less than the error of the HEMP code. This lends credibility to this method of analysis.

Figure 11 gives the results of an elastic-viscoplastic analysis for this example. Since there is damping in this type of analysis the elastic-viscoplastic analysis was stiffer than the elastic-plastic analysis.

Various techniques could be used to improve the error found in this example. Smaller time steps could be used; and this would reduce the error but increase the computation time. A smaller grid with more nodal points could be used but it must be remembered that the more degrees of freedom a problem has, the more mode shapes it will have and consequently the more chances of it going into an unstable mode shape. Also the approximations used in the large deflection analysis assume an element that has plate-like characteristics.

This is a highly efficient way to handle a very difficult problem. Its advantages are in the speed and accuracy it produces. The element equilibrium equations, although simple in their calculation of the element accelerations, allow for minor error corrections.

REFERENCES

1. O. C. Zienkiewicz, "The Finite Element Method in Engineering Science," McGraw-Hill Publishing Company, London, 1972.

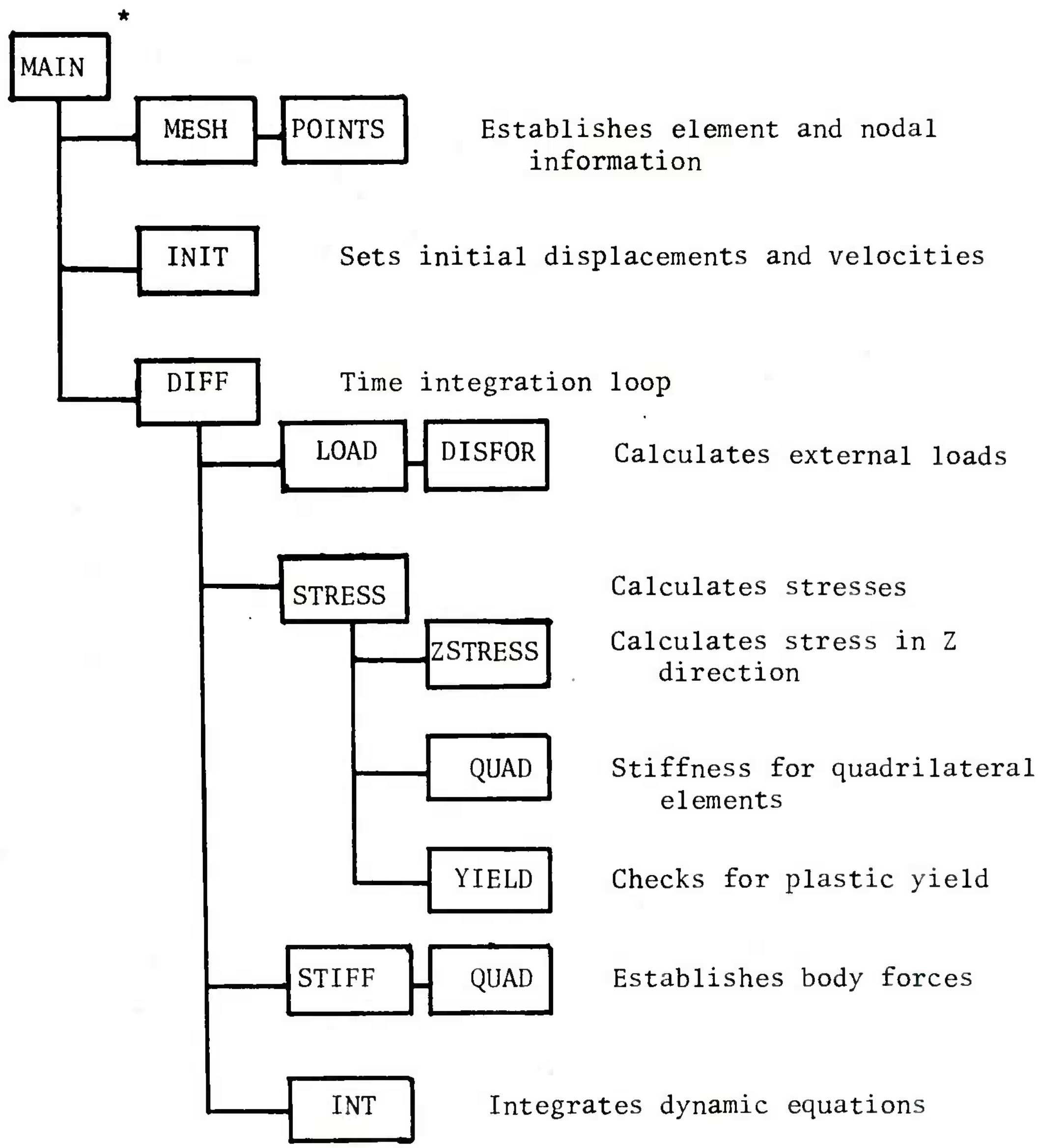
2. N. M. Newmark, "A Method of Computation for Structural Dynamics," Journal of Engineering Mechanics Division, ASCE, July 1959.

3. S. P. Chan, H. L. Cox, and W. A. Benfield, "Transient Analysis of Forced Vibrations of Complex Structural Mechanical Systems," Journal of Royal Aeronautical Society, July 1962.

4. R. Hill, "The Mathematical Theory of Plasticity," Clarendon Press, Oxford, 1950.

5. J. N. Majerus and R. R. Knapp, "Dynamic Behavior of a Multi-Layered Plate Due to an Intense Impulsive Load," Proceedings of the Second International Conference on Mechanical Behavior of Materials, Boston, August 1976.

6. Cristescu, N., "Dynamic Plasticity," North Holland Publishing Company, 1967.



*Note: Some minor subroutines which are called repeatedly are omitted for clarity.

Figure 1. Flow Diagram of TIP/I Program

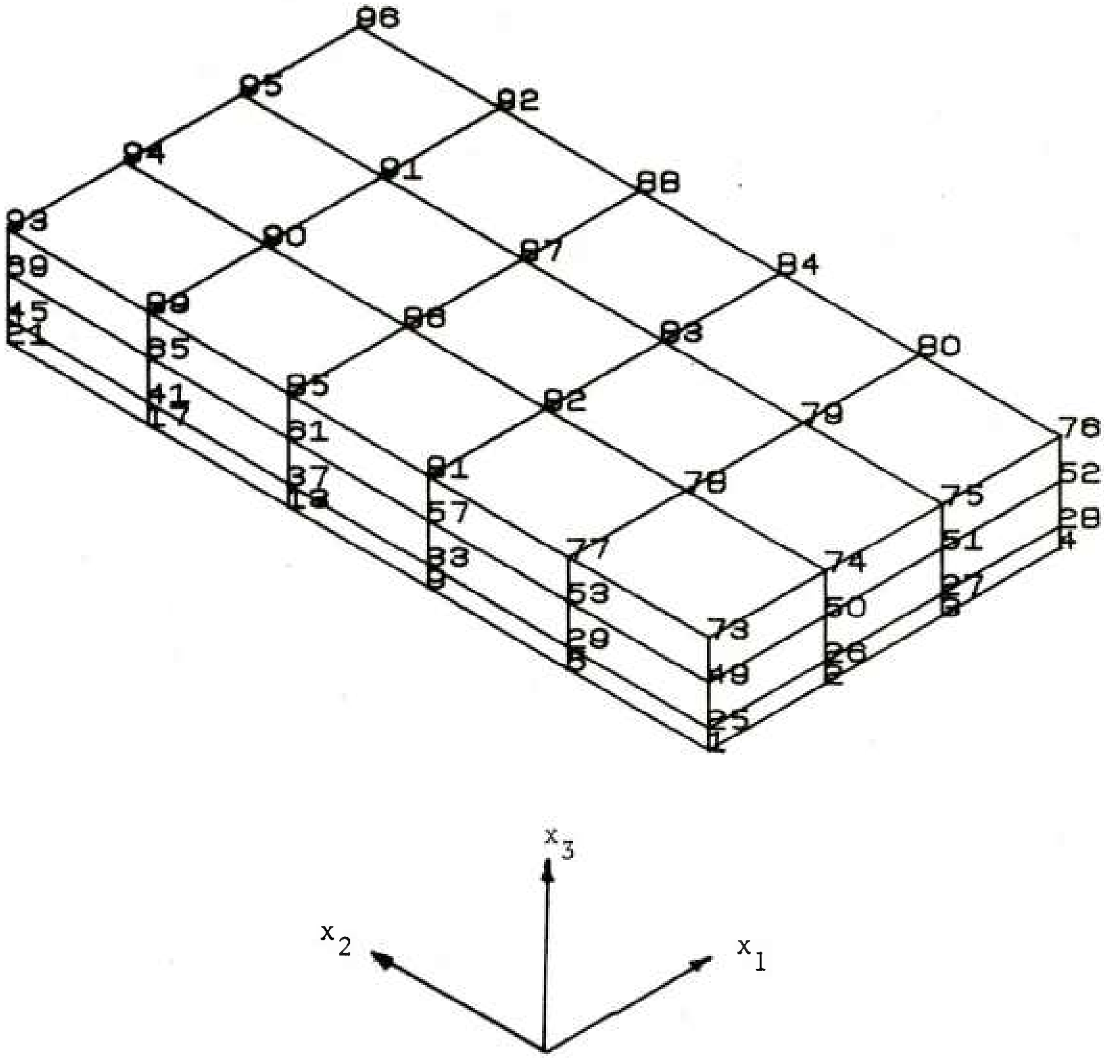
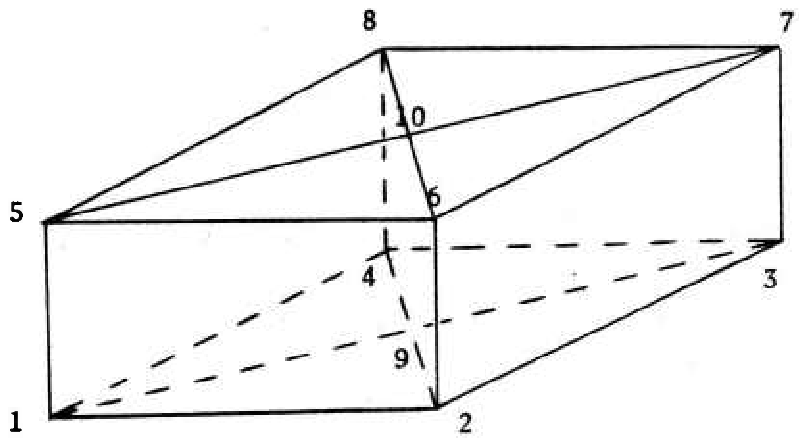


Figure 2. Coordinates and Global Numbering System.



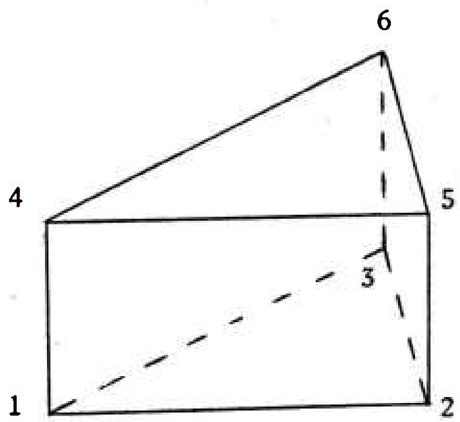


Figure 3. Nodal Numbering System for Quadrilateral and Triangular Elements.

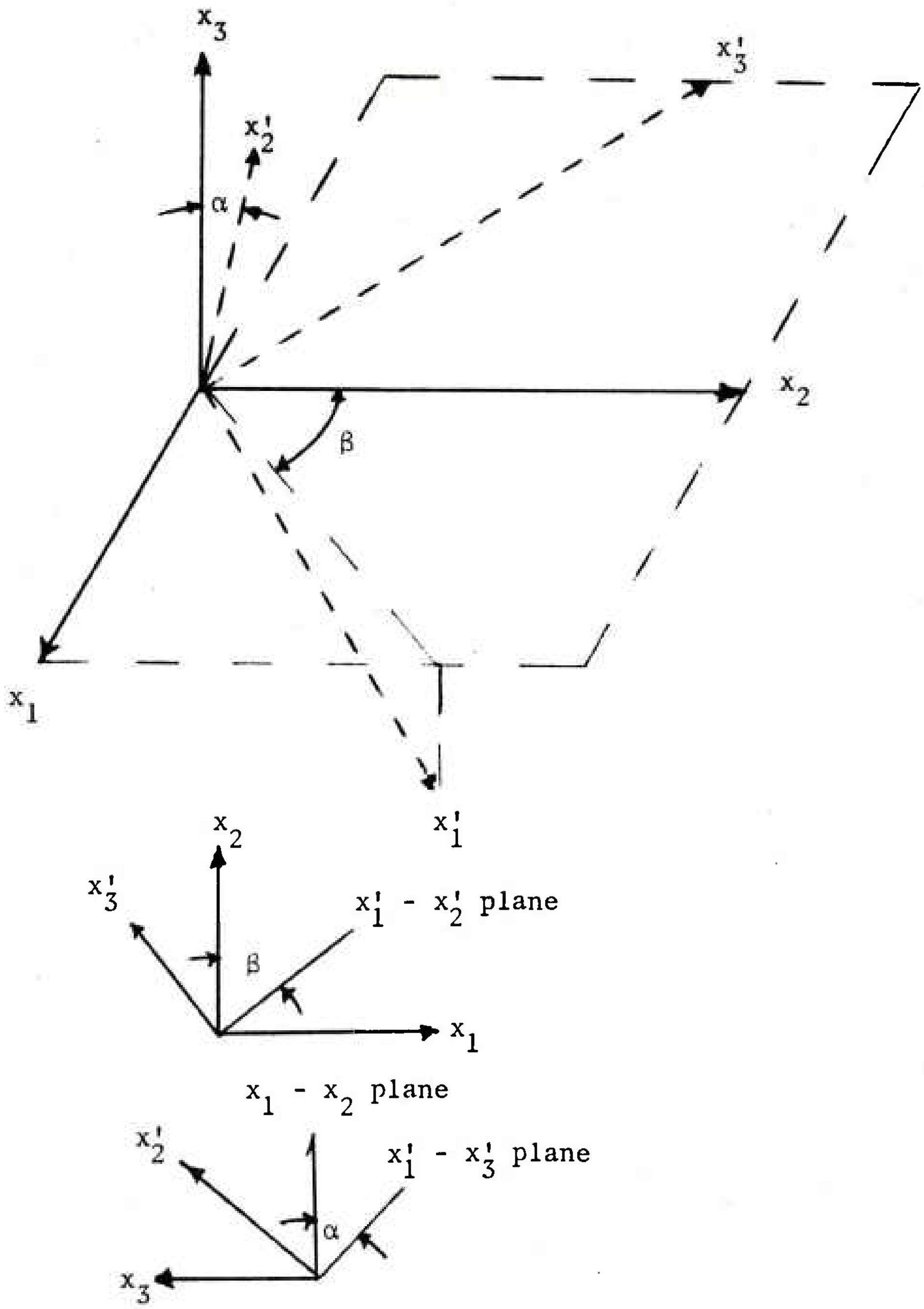


Figure 4. Relation Between Global and Orthotropic Properties.

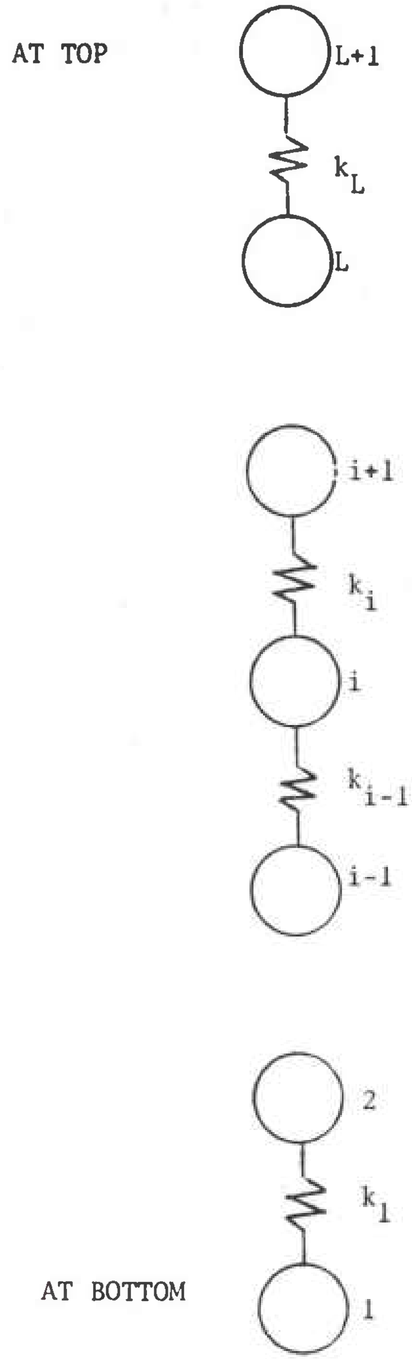


Figure 5. Model Representing the Predicted Stiffness in the Thickness Direction.

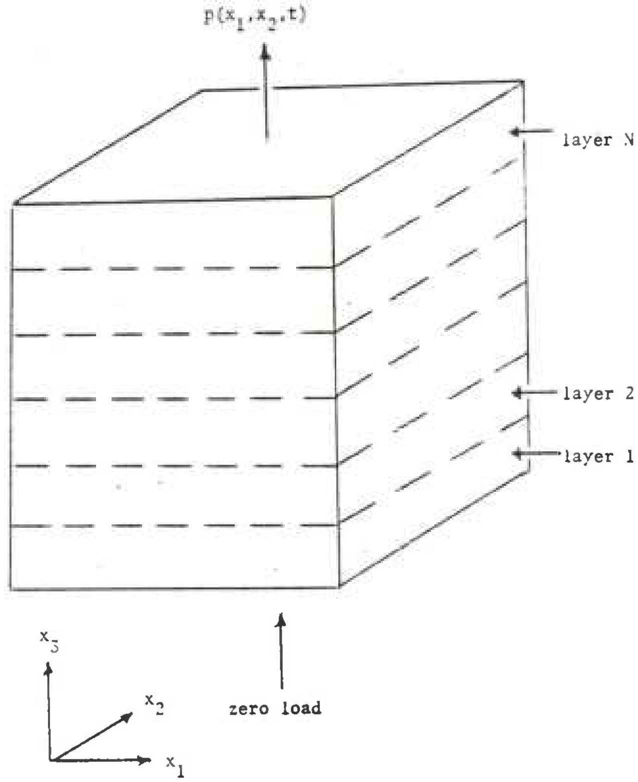


Figure 6. Section of a Laminated Plate Used in Additional Element Equilibrium Equations

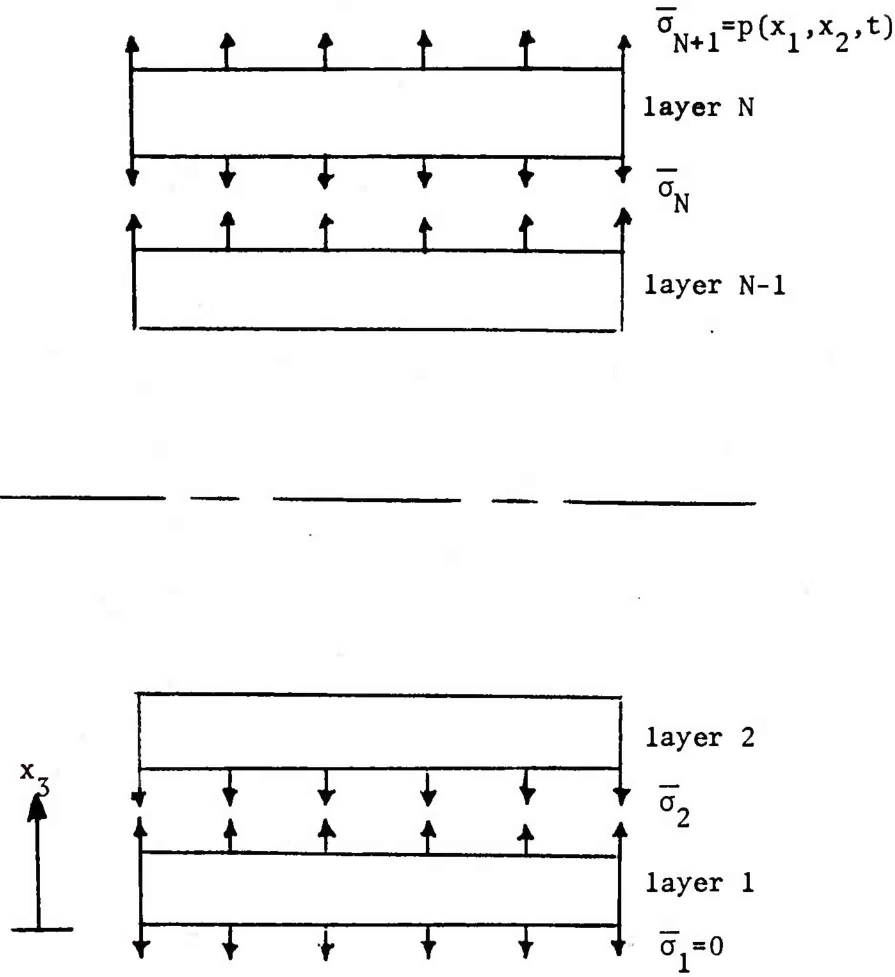


Figure 7. Definitions of Interlaminar Normal Stresses.

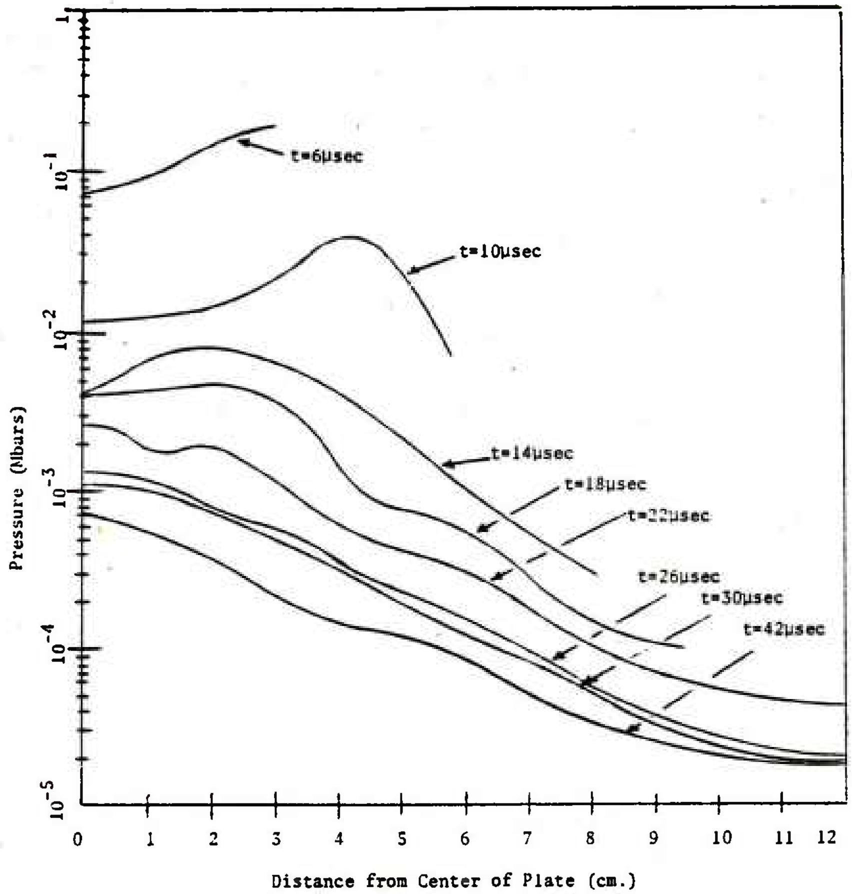


Figure 8. Actual Pressure on Plate at Discrete Time Intervals

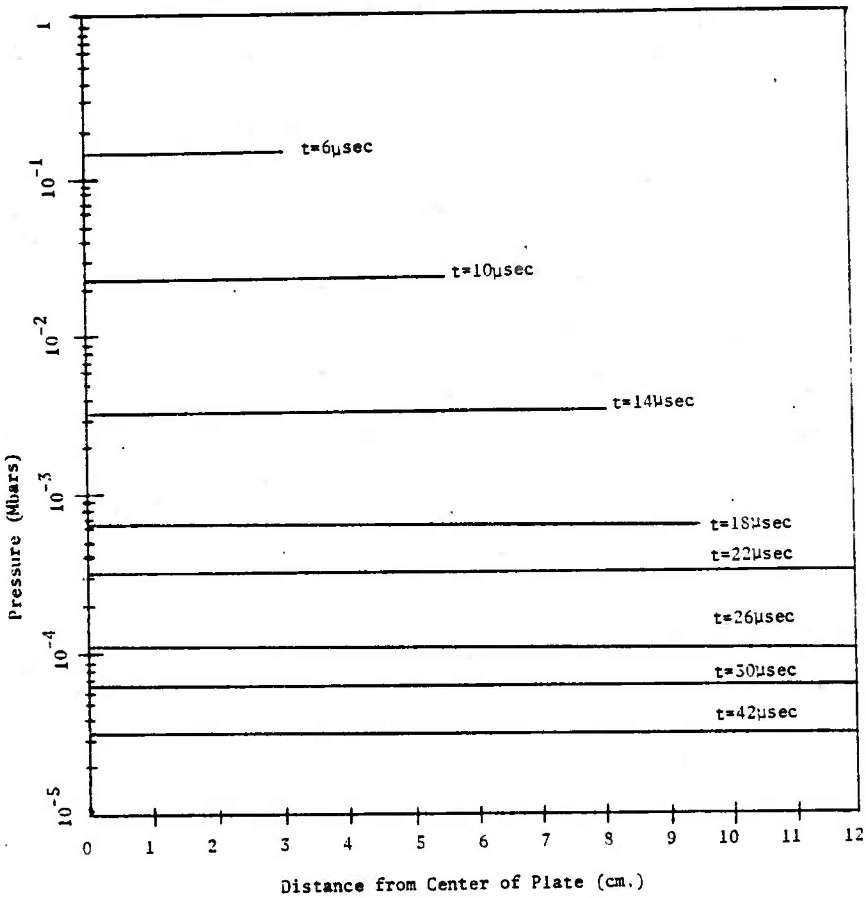


Figure 9. Idealized Pressure on Plate at Discrete Time Intervals

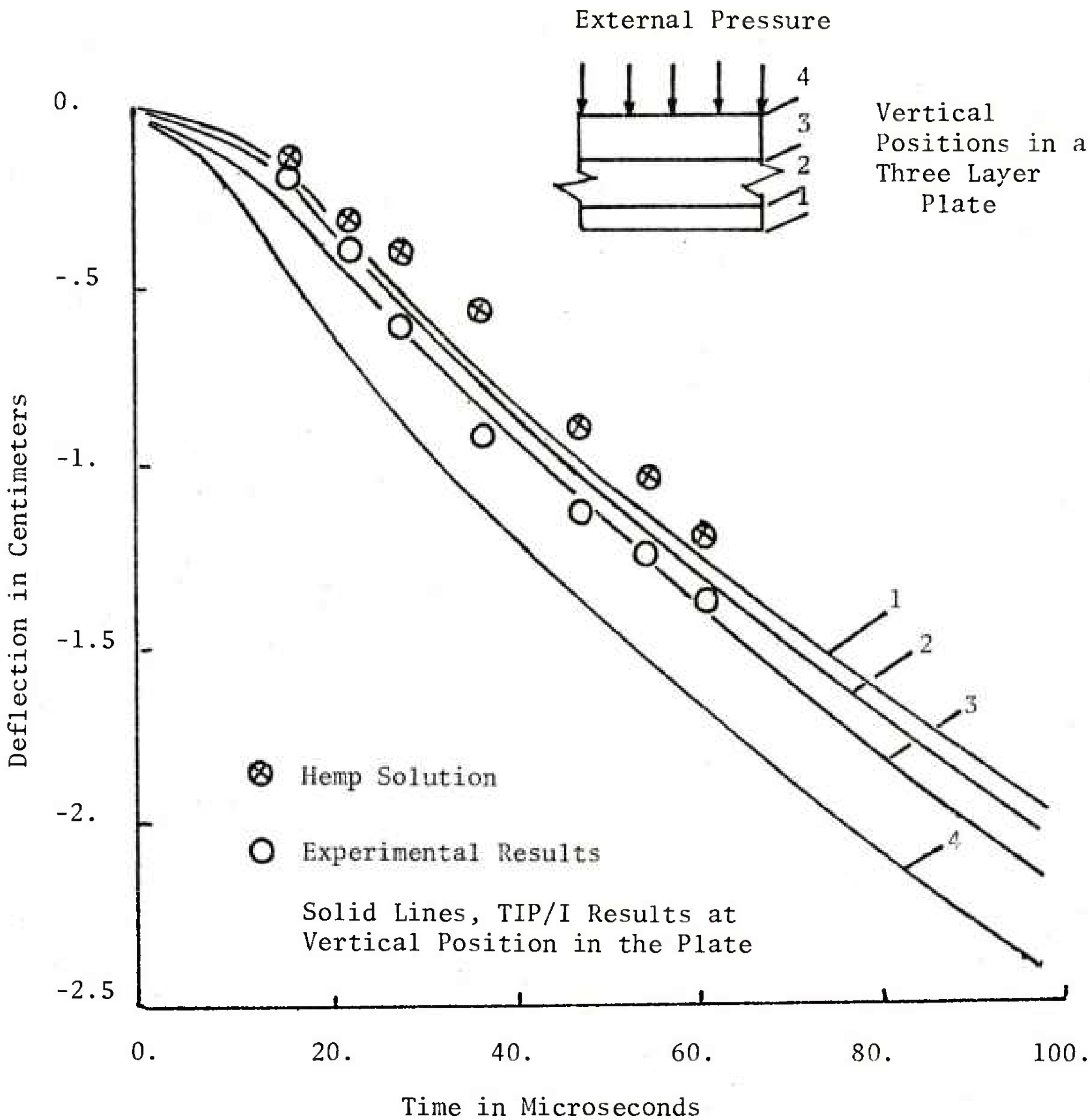


Figure 10. Comparison of Results with Three Time Increments for the Elastic-Plastic Model.

h= 2.5 μsec t= 1, 20 μsec
h= 5.0 μsec t= 20, 60 μsec
h= 10.0 μsec t= 60,100 μsec

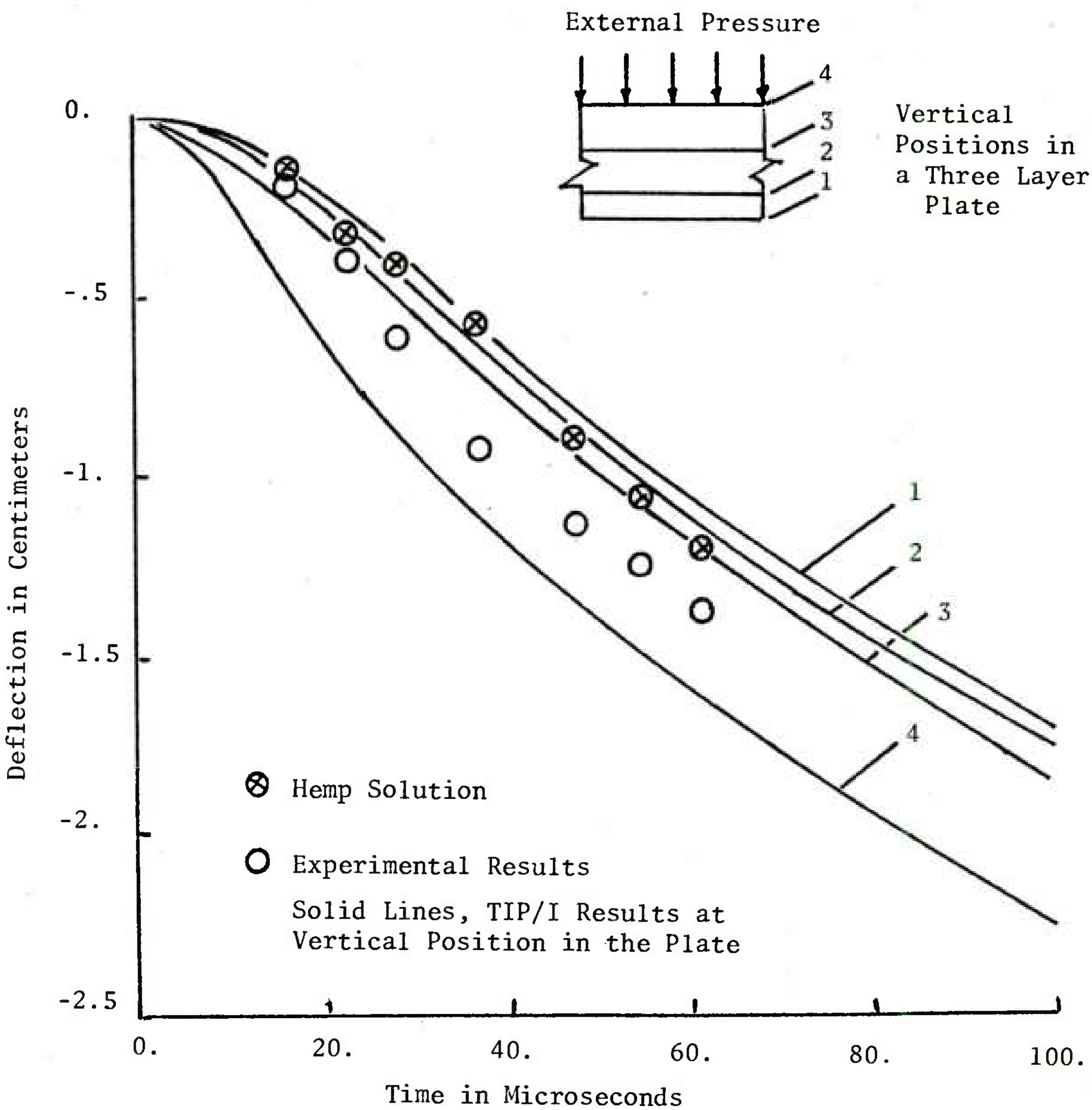


Figure 11. Comparison of Results with Three Time Increments for the Elastic-Viscoplastic Model.

h= 2.5 μsec	t= 1, 20 μsec
h= 5.0 μsec	t= 20, 60 μsec
h= 10.0 μsec	t= 60,100 μsec

APPENDIX A

COMPUTER PROGRAM INPUT CARD DESCRIPTION

TITLE CARD

Format (20A4) Title (Title for particular case)

CONTROL CARD

Format (4I5)

Columns	1-5	NUMMAT (Number of different materials; 6 maximum)
	6-10	NUMLA (Number of layers; 12 maximum)
	11-15	NLINC (Number of load increments with time; NLINC≥1)
	16-20	IPLOT (Plot parameter, 1 if plot required, 0 if no plot required)

PRINT CARD

This card controls the output that is generated so that the deflections are printed every NPRINT time increments.

Format (I5)

Columns	1-5	NPRINT

TIME INCREMENT CARD

Format (F10.5,5E10.4)

Columns	1-10	BET (β, acceleration parameter or Newmark's parameter, 0.25)
	11-20	First time step to be used
	21-20	Second time step to be used
	31-40	Third time step to be used
	41-50	Time at which program begins using second time step
	51-60	Time at which program begins using third time step

MESH GENERATION CONTROL CARD

Format (5I5)

Columns	1-5	MAXI (Maximum value of I in mesh; 25 maximum)
	5-10	MAXJ (Maximum value of J in mesh; 100 maximum)
	10-15	NSEG (Number of line segment cards)
	16-20	NBC (Number of boundary condition cards)
	21-25	NMTL (Number of material block cards)

LINE SEGMENT CARDS

The order of line segment cards is immaterial, except when plots are requested; in this case, the line segment cards must define the perimeter of solid continuously. The order of line segment cards defining internal straight lines is always irrelevant.

Format (3(2I3,2F8.3),I5)

Columns	1-3	I coordinate of 1st point
	4-6	J coordinate of 1st point
	7-14	R coordinate of 1st point
	15-22	Z coordinate of 1st point
	23-25	I coordinate of 2nd point
	26-28	J coordinate of 2nd point
	29-36	R coordinate of 2nd point
	37-44	Z coordinate of 2nd point
	45-47	I coordinate of 3rd point
	48-50	J coordinate of 3rd point
	51-58	R coordinate of 3rd point
	59-66	Z coordinate of 3rd point
	67-71	Line segment type parameter

If the number in column 71 is:

0	Point (input only 1st point)
1	Straight line (input only 1st and 2nd points)
2	Straight line as an internal diagonal (input only 1st and 2nd points)
3	Circular arc specified by 1st and 2nd points at the mid-point of the arc
4	Circular arc specified by 1st and 2nd points at the ends of the arc with the coordinates of the center of the arc given as the 3rd point (delete I and J for 3rd point)
5	Straight line as a boundary diagonal for which I of 1st point is minimum for its row and/or I of 2nd point is minimum for its row (input only 1st and 2nd points)
6	Straight line as a boundary diagonal for which I of 1st point and/or 2nd point is maximum for its row (input only 1st and 2nd points)

Note: In specifying a circular arc, the points are ordered such that a counter-clockwise direction about the center is obtained upon moving along the boundary.

BOUNDARY CONDITION CARDS

Each card assigns a boundary condition code to a block of successive nodal points starting with N1 and ending with N2, inclusive. (If code is 0 no card necessary.) Displacements and velocities can be prescribed at any node for a particular time through Subroutines BCDEL and BCVEL.

Format (2I5,I10)

Columns	1-5	Starting node number N1
	6-10	Ending node number N2
	11-20	Boundary condition code

If the number in columns 11-20 is:

- 0 displacement is not prescribed (program assigns this code automatically)
- 1 displacement is prescribed in x_1 direction
- 2 displacement is prescribed in x_2 direction
- 3 displacement is prescribed in x_3 direction
- 4 displacement is prescribed in x_1 and x_2 directions
- 5 displacement is prescribed in x_2 and x_3 directions
- 6 displacement is prescribed in x_3 and x_1 directions
- 7 displacement is prescribed in x_1, x_2 and x_3 directions
- 8 through 14, these codes parallel codes 1 through 7 except velocities are prescribed instead of displacements for chosen nodes

MATERIAL BLOCK ASSIGNMENT CARD

Each card assigns a material definition number to a block of elements defined by the I, J coordinates. Two cards for each layer.

Card 1

Format (I5,3F10.0)

Columns	1-5	Material definition number (1 through 6)
	6-15	Material principal property inclination angle BETA in x_1 - x_2 plane
	16-25	Material principal property inclination angle ALPHA in x'_1 - x'_2 plane
	26-35	Bingham viscosity in this layer

Card 2**

Format (6E12.6)

Columns	1-12	Yield stress in x'_1 direction
	13-24	Yield stress in x'_2 direction
	25-36	Yield stress in x'_3 direction
	37-48	Yield stress in x'_1 x'_2 direction
	49-60	Yield stress in x'_2 x'_3 direction
	61-72	Yield stress in x'_1 x'_3 direction

** Note: If material is isotropic, place yield stress in first 12 spaces and leave the rest blank. When the program finds blanks in the 13-24 spaces it assumes the material is isotropic and calculates the other yield stresses.

MATERIAL PROPERTY INFORMATION CARDS

The following group of cards must be specified for each material (Maximum of 6).

a. MATERIAL IDENTIFICATION CARD

Format (I5,F10.0)

Columns	1-5	Material identification number
	6-15	Mass density of material (if required)

b. MATERIAL PROPERTY CARDS

First Card

Format (6F10.0)

Columns	1-10	Modulus of elasticity, E_1
	11-20	Modulus of elasticity, E_2
	21-30	Modulus of elasticity, E_3
	31-40	Poisson's ratio, ν_{12}
	41-50	Poisson's ratio, ν_{23}
	51-60	Poisson's ratio, ν_{31}

Second Card

Format (3F10.0)

Columns	1-10	Shear Modulus, G_{12}
	11-20	Shear Modulus, G_{23}
	21-30	Shear Modulus, G_{31}

LAYER THICKNESS CARD

Format (12F5.3)

Columns	1-5	TH(1) (Thickness of layer 1)
	6-10	TH(2) (Thickness of layer 2)
	11-15	TH(3) (Thickness of layer 3)
	etc. up to TH (NUMLA)	

PLOT TITLE CARD*

Format (8A10)

Columns	1-80	Title (Title printed under each plot)

PLOT GENERATION INFORMATION CARD*

Format (2F10.0)

Columns	1-10	RMAX (Maximum x_1 coordinate of mesh)
	11-20	ZXAX (Maximum x_2 coordinate of mesh)

*Note: use only if IPLOT = 1 (plot required)

INITIAL CONDITION CARDS

Card 1

Format (I5)

Columns	1-5	INIDV number of initial displacement and velocities cards (see Card 2) specifying initial displacements and velocities

Card 2

Each card assigns an initial displacement and/or initial velocity to a specific nodal point.** The number of these cards is equal to INIDIV.

Format (I5,2E12.6)

Columns	1-5	Nodal Point
	6-17	Initial Displacement
	18-29	Initial Velocity

**Note: Used only if there are initial displacements or velocities, otherwise the program initializes these values as zero and no input is necessary.

APPENDIX B

PROGRAM TIP FOLLOWED BY ITS INPUT

```
      PROGRAM TIPI(INPUT,OUTPUT,DATP,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1,
     1TAPE2,TAPE3,TAPE4=DATP)
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C THREE DIMENSIONAL ANALYSIS OF AN IMPULSIVELY LOADED LAMINATED PLATE
C BY THE FINITE ELEMENT METHOD WITH ORTHOTROPIC PLASTIC YIELDING
C AND NONLINEAR STRAINS
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      INTEGER CODE
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA,NCG
      COMMON/NPR/NPRINT
      COMMON/MATP/RO(12),E(9,12),EE(9),ETA(12)
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/NPDATA/X(200),Y(200),CODE(200),NPNUM(10,20)
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
      COMMON/RESULT/D(6,6),C(6,6),CNS(6,6)
      COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
      COMMON/BASIC2/BET,H,HN,HH(3),HT(2),TIME,NLAY,IFLAG,IT,NTIME
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
      COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
      COMMON/MASS/A(600),B(600),CM(8)
      COMMON/FIRST/FO(600),DER(600)
      COMMON/DELT/XDEL(4,18)
      COMMON/DELTRI/DELTA(4,18)
      COMMON/PLYLD/SIGY(12,6),DEPS(6),EPS(12,50,6)
      DIMENSION TITLE(20)
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     READ AND WRITE CONTROL INFORMATION
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
   50 READ(5,1000)TITLE,NUMMAT,NUMLA,NLINC,IPLOT
      IF(EOF(5))920,88
   88 READ(5,1004)NPRINT
      IF(EOF(5).NE.0.)GO TO 920
      WRITE(6,2000)TITLE,NUMLA,NUMMAT,NLINC
      WRITE(4)NLINC,NUMLA
      READ(5,1002) BET,(HH(I),I=1,3),(HT(J),J=1,2)
      WRITE(6,2001)BET,(HH(I),I=1,3),(HT(J),J=1,2)
      H=HH(1)
      HN=0.0
      TIME=0.00
      NTIME=0
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     GENERATE FINITE ELEMENT MESH
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
  100 CALL MESH
      MPRINT=0
      DO 230 N=1,NUMNP
      IF(MPRINT.NE.0) GO TO 220
      WRITE(6,2003)
      MPRINT=59
```

```
  220 MPRINT=MPRINT-1
  230 WRITE(6,2004) N,X(N),Y(N)
      IT=NUMNP*(NUMLA+1)*3
  440 MPRINT=0
      DO 460 N=1,NUMEL
      IF(MPRINT.NE.0) GO TO 450
      WRITE(6,2008)
      MPRINT=59
  450 MPRINT=MPRINT-1
      II=IX(N,1)
      JJ=IX(N,2)
      KK=IX(N,3)
      LL=IX(N,4)
  460 WRITE(6,2009) N,(IX(N,I),I=1,4)
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     READ AND WRITE MATERIAL PROPERTIES
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
  500 CONTINUE
      DO 510 M=1,NUMMAT
      READ(5,1004)  MTYPE,(RO(MTYPE))
      WRITE(6,2010) MTYPE,   RO(MTYPE)
      READ(5,1005)(E(J,MTYPE),J=1,9)
      WRITE(6,2011)(E(J,MTYPE),J=1,9)
 510  CONTINUE
      READ(5,1006)(TH(I),I=1,NUMLA)
      WRITE(6,1007)
 1007 FORMAT("OTHICKNESSES")
      WRITE(6,1006)(TH(I),I=1,NUMLA)
      WRITE(6,1008)
 1008 FORMAT("OYIELD STRESSES")
      WRITE(6,1009)((SIGY(I,J),J=1,6),I=1,NUMLA)
 1009 FORMAT(6(2X,E15.7))
      WRITE(6,1108)(ETA(I),I=1,NUMLA)
 1108 FORMAT(//" COEFFICIENTS OF VISCOSITY"/6(2X,E12.6)/)
      CALL USTART
      CALL UDIMEN(10.,10.)
      IF (IPLOT.EQ.1) CALL MPLOT
      DO 800 I=1,NUMLA
      DO 800 J=1,NUMEL
      DO 800 K=1,6
      SIGMA(I,J,K)=0.00
  800 CONTINUE
      CALL INIT
      DO 900 NL=1,NLINC
      IF(NL.GT.1) GO TO 721
      DO 720 N=1,NUMLA
      ALPHA(N)=ALPHA(N)/57.295780
  720 BETA(N)=BETA(N)/57.295780
  721 CONTINUE
C
```

```
C     FORM STIFFNESS MATRIX
C
      DO 850 I=1,4
      DO 850 J=1,18
      DELTA(I,J)=0.00
  850 CONTINUE
      CALL DIFF
  900 CONTINUE
      CALL UEND
  910 GO TO 50
 1000 FORMAT(20A4/6I5,F5.0,5I5)
 1001 FORMAT(3F10.0)
 1002 FORMAT(F10.5,5E10.4)
 1004 FORMAT (I5,F10.0)
 1005 FORMAT(6F10.0)
 1006 FORMAT (12F5.3)
 2000 FORMAT (2H1 ,20A4/
     1 33H0  NUMBER OF LAYERS---------------I4/
     2 33H0  NUMBER OF MATERIALS------------I4/
     3 33H0  NUMBER OF LOAD INCREMENTS------I4/)
 2001 FORMAT(41H0  ACCELERATION PARAMETER, BETA----------F10.5/
     148H0  TIME-STEP SIZE,HH-----------------------------3(2X,E10.4)/
     251H0  TIME TO CHANGE TIME STEP SIZE-------------------2(2X,E10.4)
     3 /)
 2003 FORMAT (35H1   N       X          Y               )
 2004 FORMAT (I5,2F10.4)
 2008 FORMAT (51H1  EL   I   J   K  L        ANGLE BETA   ANGLE ALPHA)
 2009 FORMAT (I5,4I4,2F13.3)
 2010 FORMAT (1H1,"MATERIAL IDENTIFICATION NUMBER =",I2/
     21H ,"MASS DENSITY =",E15.7)
 2011 FORMAT (
     11H ,"MODULUS OF ELASTICITY-EN =",E15.7/
     21H ,"MODULUS OF ELASTICITY-ES =",E15.7/
     31H ,"MODULUS OF ELASTICITY-ET =",E15.7/
     41H ,"POISSON RATIO-NUNS =",E15.7/
     51H ,"POISSON RATIO-NUNT =",E15.7/
     61H ,"POISSON RATIO-NUST =",E15.7/
     71H ,"SHEAR MODULUS-GNS =",E15.7/
     81H ,"SHEAR MODULUS-GST =",E15.7/
     91H ,"SHEAR MODULUS-GTN =",E15.7/)
 2016 FORMAT (26H  THE SYSTEM CONVERGED IN I2,11H ITERATIONS)
 2017 FORMAT (33H  THE SYSTEM DID NOT CONVERGE IN I2,11H ITERATIONS)
  920 STOP
      END
      SUBROUTINE ANGLE (R,Z,RC,ZC,ANG)
      INTEGER CODE
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA,NCG
      COMMON/MATP/RO(12),E(9,12),EE(9),ETA(12)
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
```

```
      COMMON/NPDATA/X(200),Y(200),CODE(200),NPNUM(10,20)
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
      COMMON/RESULT/D(6,6),C(6,6),CNS(6,6)
      COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
      COMMON/BASIC2/BET,H,HN,HH(3),HT(2),TIME,NLAY,IFLAG,IT,NTIME
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,5),DSIG(6),F(600)
      COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
      COMMON/MASS/A(600),B(600),CM(8)
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     FIND ANGLE OF INCLINATION BETWEEN 0 AND 2*PI
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      PI=3.1415927
      D1=(Z-ZC)
      D2=(R-RC)
      IF(ABS(R-RC).GT.1.E-8) GO TO 100
      ANG=PI/2.
      IF(D1.GT.1.E-8) RETURN
      ANG=-ANG
      RETURN
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     ALLOW CIRCLE TO CROSS AXIS
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
  100 ANG=ATAN2(D1,D2)
      RETURN
      END
      SUBROUTINE BAND (A,B,NROW,NCOL,NMID,NRHS,KO)
      DIMENSION A(13,3),B(13,1)
      NL=NMID-1
      NR=NCOL-NMID
      NRM=NROW-1
      DO 2 I=1,NRM
      IF(A(I,NMID).EQ.0.0) GO TO 7
      RE=1.0/A(I,NMID)
      LL=NROW-I
      LR=LL
      IF(LL.GT.NL) LL=NL
      IF(LR.GT.NR) LR=NR
      DO 2 J=1,LL
      JR=I+J
      JC=NMID-J
      RA=-A(JR,JC)*RE
      DO 1 K=1,LR
      KC=JC+K
      IC=NMID+K
    1 A(JR,KC)=A(JR,KC)+RA*A(I,IC)
      DO 2 K=1,NRHS
    2 B(JR,K)=B(JR,K)+RA*B(I,K)
      IF(A(NROW,NMID).EQ.0.0) GO TO 6
      RE=1.0/A(NROW,NMID)
      DO 3 I=1,NRHS
```

```
    3 B(NROW,I)=B(NROW,I)*RE
      DO 5 I=1,NRM
      IR=NROW-I
      RE=1./A(IR,NMID)
      LR=I
      IF(LR.GT.NR) LR=NR
      DO 4 J=1,LR
      JR=IR+J
      JC=NMID+J
      DO 4 K=1,NRHS
    4 B(IR,K)=B(IR,K)-A(IR,JC)*B(JR,K)
      DO 5 J=1,NRHS
    5 B(IR,J)=B(IR,J)*RE
      KO=0
      RETURN
    6 I=NROW
    7 KO=I
      WRITE(6,100) KO
      RETURN
  100 FORMAT(//" YOU GOOFED - - - THERE IS A ZERO ON THE PRINCIPAL DIAGO
     1NAL IN THE",I4," TH ROW."//)
      END
      SUBROUTINE BCDEL(I)
      COMMON/BASIC2/BET,H,HN,HH(3),HT(2),TIME,NLAY,IFLAG,IT,NTIME
      COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C IF DISPLACEMENTS ARE GIVEN AT SPECIFIED NODES, THIS IS THE POINT IN
C THE SUBROUTINE WHERE THAT SPECIFIC INPUT IS GIVEN WITH  RESPECT  TO
C TIME AND NODAL LOCATION
C * * * * * * * * * * * * * * * * * * * *  * * * * * * * * * * * * * *
      DEL(I)=0.0
      RETURN
      END
      SUBROUTINE BCVEL(I)
      COMMON/BASIC2/BET,H,HN,HH(3),HT(2),TIME,NLAY,IFLAG,IT,NTIME
      COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
      COMMON/FIRST/FO(600),DER(600)
      COMMON/MASS1/XMINV(600),EEKM(13),F1(600),F2(600)
      DERN=DER(I)
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C IF VELOCITIES ARE GIVEN AT SPECIFIED NODES, THIS IS THE POINT IN THE
C SUBROUTINE WHERE THAT SPECIFIC INPUT IS GIVEN WITH RESPECT  TO  TIME
C AND NODAL LOCATION
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DER(I)=0.0
      DEL(I)=DELN(I)+(1.-2.*BET)*H*DERN+2.*BET*H*DER(I)+(.5-2.*BET)
     1 *H**2*XMINV(I)*(GNM1(I)+F1(I))
      RETURN
      END
      SUBROUTINE CIRCLE(ANG1,DELPHI,RSTRT,ZSTRT,RC,ZC,I,J)
```

```
      INTEGER CODE
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA,NCG
      COMMON/MATP/RO(12),E(9,12),EE(9),ETA(12)
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/NPDATA/X(200),Y(200),CODE(200),NPNUM(10,20)
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
      COMMON/RESULT/D(6,6),C(6,6),CNS(6,6)
      COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
      COMMON/BASIC2/BET,H,HN,HH(3),HT(2),TIME,NLAY,IFLAG,IT,NTIME
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
      COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
      COMMON/MASS/A(600),B(600),CM(8)
      DIMENSION AR(10, 40),AZ(10, 40)
      EQUIVALENCE (X(1),AR),(Y(1),AZ)
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     FIND INTERSECTION OF LINE AND CIRCLE = NEW R AND Z
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      ANG1=ANG1+DELPHI
      RR=SQRT((RSTRT-RC)**2+(ZSTRT-ZC)**2)
      AR(I,J)=RC+RR*COS(ANG1)
      AZ(I,J)=ZC+RR*SIN(ANG1)
      RETURN
      END
      SUBROUTINE DIFF
      INTEGER CODE
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA,NCG
      COMMON/MATP/RO(12),E(9,12),EE(9),ETA(12)
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/NPDATA/X(200),Y(200),CODE(200),NPNUM(10,20)
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
      COMMON/RESULT/D(6,6),C(6,6),CNS(6,6)
      COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
      COMMON/BASIC2/BET,H,HN,HH(3),HT(2),TIME,NLAY,IFLAG,IT,NTIME
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
      COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
      COMMON/MASS/A(600),B(600),CM(8)
      COMMON/FIRST/FO(600),DER(600)
      COMMON/MASS1/XMINV(600),EEKM(13),F1(600),F2(600)
      COMMON/NPR/NPRINT
      COMMON/PLYLD/SIGY(12,6),DEPS(6),EPS(12,50,6)
      HN=H
      TIME=TIME+H
      NTIME=NTIME+1
      DO 20 I=1,2
      IF(TIME.LT.HT(I).OR.H.EQ.HH(I+1))GO TO 20
      IF(TIME.GT.HT(2).AND.H.EQ.HH(3))GO TO 30
      H=HH(I+1)
      TIME=TIME-HN+H
```

```
      WRITE(6,10)HN,H,TIME,HT(I)
   10 FORMAT("0 TIMES ",4(2X,E10.4)/)
      GO TO 30
   20 CONTINUE
   30 IF(NTIME.GT.1)GO TO 280
      WRITE(6,999)
  999 FORMAT("1INITIAL FORCE VECTOR"/)
      WRITE(6,1000)(FO(I),I=1,IT)
      WRITE(6,998)
  998 FORMAT("0INITIAL DISPLACEMENTS")
      WRITE(6,1000)(DEL(I),I=1,IT)
      WRITE(6,997)
  997 FORMAT("0INITIAL VELOCITIES")
      WRITE(6,1000)(DER(I),I=1,IT)
      WRITE(6,996)
  996 FORMAT("1")
 1000 FORMAT(9(2X,E12.6))
  280 CONTINUE
      DO 390 I=1,IT
      IF(NTIME.EQ.1)GO TO 390
      F2(I)=F1(I)
  390 F1(I)=F(I)
      CALL LOAD
      CALL STRESS
      CALL STIFF
      CALL INT
      IF(NTIME.EQ.1) GO TO 98
      IF((NTIME/NPRINT)*NPRINT.NE.NTIME) GO TO 99
   98 CONTINUE
C     WRITE(6,26)((I,J,(EPS(I,J,K),K=1,6),I=1,NUMLA),J=1,NUMEL)
C     WRITE(6,25)((I,J,(SIGMA(I,J,K),K=1,6),I=1,NUMLA),J=1,NUMEL)
      WRITE(6,1003)NTIME,TIME
 1003 FORMAT(/"0DEFLECTION FOR CYCLE=",I3,"    TIME =",E12.6/
     1 "  NODE",15X,"DEL X",15X,"DEL Y",15X,"DEL Z"/)
      NUP=NUMLA+1
      DO 1026 K=1,NUP
      DO 1026 I=1,NUMNP
      NP=I+(K-1)*NUMNP
      NP3=3*NP
      WRITE(6,1001)NP,DEL(NP3-2),DEL(NP3-2),DEL(NP3)
 1001 FORMAT(I6,3E20.10)
 1026 CONTINUE
      MI=NUMNP*3
C     WRITE(6,15)(MX,DEL(MX),DELN(MX),DELN1(MX),B(MX),GNM1(MX),
C    1 GNM2(MX),F(MX),MX=3,IT,MI)
   15 FORMAT(/"   #",5X,"DEL N",8X,"DELN N-1",5X,"DELN1 N-2",8X,
     1 "B  N",8X,"GNM1 N-1",8X,"GNM2 N-2",8X,"F"
     2 /(I4,2X,E12.6,2X,E12.6,2X,E12.6,2X,E12.6,2X,E12.6,2X,E12.6,
     3 2X,E12.6))
   25 FORMAT(//"1   LAYER",2X,"ELEMENT",5X,"SIGMA X ",6X,"SIGMA Y ",6X,
```

```
     1 "SIGMA Z ",6X,"SIGMA XY",6X,"SIGMA YZ",6X,"SIGMA XZ"/(2I8,2X,6E14
     2 .6))
  26 FORMAT(//"    LAYER",2X,"ELEMENT",5X,"  EPS X ",6X,"  EPS Y ",6X,
     1 "  EPS Z ",6X,"  EPS XY",6X,"  EPS YZ",6X,"  EPS XZ"/(2I8,2X,6E14
     2 .6))
  99 CONTINUE
C TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
      NTOP=NUMLA*3*NUMNP+3
      ETIM=TIME*1.0E+06
      NL1=3*NUMNP
      NL2=(NUMLA+1)*NUMNP*3
      WRITE(4)DEL(3),DEL(NTOP),(DEL(IJ),IJ=4,NL2,NL1),ETIM
C TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
      IF((NTIME/5)*5.NE.NTIME)GO TO 1039
      CALL MPLOT
 1039 CONTINUE
      RETURN
      END
      SUBROUTINE DISFOR
      COMMON/BASIC2/BET,H,HN,HH(3),HT(2),TIME,NLAY,IFLAG,IT,NTIME
      COMMON/FOR/FF(5),FC(5)
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/NUMB/NITER,HDAT,TO
      DIMENSION DIST(5)
      DIMENSION FF1(19)
      DATA (FF1(I),I=1,19)/1.4628564E-1,8.561575E-2,2.2316769E-2,
     1 6.0124041E-3,2.2297704E-3,1.1280587E-3,6.0944142E-4,
     1 3.5506404E-4,2.2665571E-4,1.5336139E-4,1.0736918E-4,
     1 8.1057611E-5,6.5648474E-5,5.5056944E-5,5.0003866E-5,
     1 4.4950788E-5,3.9897700E-5,3.6072003E-5,3.2246295E-5/
  11  NITER=19
      TNITER=42.0E-6
      HDAT=2.0E-6
      TO=6.0E-6
      DO 10 I=1,5
      DIST(I)=SQRT(XXX(I)**2+YYY(I)**2)
   10 CONTINUE
      YBLAST=.2362*(TIME*1.0E+06)
      IF(TIME.GT.TO) GO TO 40
      DO 45 I=1,4
   45 FF(I)=-FF1(1)*14.5E6
      GO TO 18
   40 IF(TIME-TNITER) 15,16,16
  15  CALL POINT(TIME,NUM,TN)
      DO 12 I=1,4
      FF(I)=-(FF1(NUM)+(FF1(NUM+1)-FF1(NUM))*(TIME-TN)/HDAT)*14.5E6
  12  CONTINUE
      GO TO 18
  16  DO 17 I=1,4
```

```
      FF(I)=-(FF1(NITER)+(FF1(NITER)-FF1(NITER-1))*(TIME-TNITER)/HDAT)
     1 *14.5E6
      IF(FF(I).GT.0.0) FF(I)=0.0
   17 CONTINUE
   18 CONTINUE
      FF(5)=(FF(1)+FF(2)+FF(3)+FF(4))/4.0
      DO 30 I=1,5
      IF(DIST(I).GT.YBLAST) FF(I)=0.0
   30 CONTINUE
      RETURN
      END
      SUBROUTINE INIT
      INTEGER CODE
      COMMON/MASS1/XMINV(600),EEKM(13),F1(600),F2(600)
      COMMON/FIRST/FO(600),DER(600)
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA,NCG
      COMMON/MATP/RO(12),E(9,12),EE(9),ETA(12)
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/NPDATA/X(200),Y(200),CODE(200),NPNUM(10,20)
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
      COMMON/RESULT/D(6,6),C(6,6),CNS(6,6)
      COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
      COMMON/BASIC2/BET,H,HN,HH(3),HT(2),TIME,NLAY,IFLAG,IT,NTIME
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
      COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
      COMMON/MASS/A(600),B(600),CM(8)
      COMMON/PLYLD/SIGY(12,6),DEPS(6),EPS(12,50,6)
C LET INITIAL DEFLECTION, VELOCITY, AND STRESS BE ZERO
C LET INITIAL FORCE BE LINC
      DO 100 I=1,IT
      DELN(I)=0.00
      DER(I)=0.00
      DEL(I)=0.0
      F1(I)=0.0
      F2(I)=0.0
      DELN1(I)=0.0
      F(I)=0.0
      B(I)=0.0
      GNM1(I)=0.00
      GNM2(I)=0.0
      FO(I)=0.00
      A(I)=0.0
  100 CONTINUE
      DO 110 I=1,NUMLA
      DO 110 J=1,NUMEL
      DO 110 K=1,6
  110 EPS(I,J,K)=0.0
      READ(5,200)INIDV
      IF(EOF(5).NE.0.0)GO TO 220
```

```
 200 FORMAT(2I5)
     READ(5,210)((ND,DEL(ND),DER(ND)),I=1,INIDV)
 210 FORMAT(I5,2E12.6)
 220 CONTINUE
     RETURN
     END
     SUBROUTINE INT
     INTEGER CODE
     COMMON/FIRST/FO(600),DER(600)
     COMMON/MASS1/XMINV(600),EEKM(13),F1(600),F2(600)
     COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA,NCG
     COMMON/MATP/RO(12),E(9,12),EE(9),ETA(12)
     COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
    1CRZ(6,6),XI(10),SIG(12),N,M
     COMMON/NPDATA/X(200),Y(200),CODE(200),NPNUM(10,20)
     COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
     COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
     COMMON/BASIC2/BET,H,HN,HH(3),HT(2),TIME,NLAY,IFLAG,IT,NTIME
     COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
     COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
     COMMON/MASS/A(600),B(600),CM(8)
     COMMON/DISP1/TF(39,4),TFI(4,4)
     DIMENSION THT(25),AOT(25),BOT(4),FOT(4),DERT(4),DELNT(4),AU(13,3),
    1 AFT(25,4),GNM1T(4),GNM2T(4),SM(4),DELN1T(4),DELT(4),BU(13,1)
    1 ,EKM(13)
     BH=BET *H**2
     DIF1=0.50-BET
     BHN=BET*H*HN
C TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
     IF(NTIME.GT.1)GO TO 789
     EMIT=0.0
     DHET=0.0
     NTOP=NUMLA*3*NUMNP+3
     NL1=NUMNP*3
     NL2=NL1*(NUMLA+1)
     WRITE(4)DEL(3),DEL(NTOP),(DEL(IJ),IJ=4,NL2,NL1),EMIT
 789 CONTINUE
C TTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
C H AND BETA ARE INPUT VARIABLES STORED IN COMMON
     NUP=NUMLA+1
     NPT=(NUMLA+1)*NUMNP
     DO 100 I=1,IT
     DELN1(I)=DELN(I)
     DELN(I)=DEL(I)
 100 CONTINUE
     IF(NTIME.GT.1)GO TO 180
C OBTAIN INVERSE OF MASS MATRIX,XMINV
     DO 120 I=1,IT
     XMINV(I)=1/A(I)
 120 CONTINUE
```

```
C STARTING PROCEDURE
C DELN AND DER ARE INITIAL CONDITIONS ON DISPLACEMENT
C GNM1 IS INPUT VECTOR
      DO 160 K=1,NPT
      ID=CODE(K)
      I=3*K
      IF(ID.EQ.0)GO TO 140
      IF(ID.EQ.3.OR.ID.EQ.5.OR.ID.EQ.6.OR.ID.EQ.7) GO TO 125
      IF(ID.EQ.10.OR.ID.EQ.12.OR.ID.EQ.13.OR.ID.EQ.14) GO TO 130
      GO TO 140
  125 CALL BCDEL(I)
      GO TO 160
  130 CALL BCVEL(I)
      GO TO 160
  140 DEL(I)=DELN(I)+H*DER(I)
      DEL(I)=DEL(I)+XMINV(I)*(BH*B(I)-DIF1*H**2*GNM1(I))
      DEL(I)=DEL(I)+XMINV(I)*(DIF1*H**2*FO(I)+BH*F(I))
  160 CONTINUE
      GO TO 240
  180 CONTINUE
      KK=0
C CALCULATE DEL BY FINITE DIFFERENCE EQUATIONS
      DO 220 K=1,NPT
      ID=CODE(K)
      I=3*K
      IF(ID.EQ.0) GO TO 200
      IF(ID.EQ.3.OR.ID.EQ.5.OR.ID.EQ.6.OR.ID.EQ.7) GO TO 185
      IF(ID.EQ.10.OR.ID.EQ.12.OR.ID.EQ.13.OR.ID.EQ.14) GO TO 190
      GO TO 200
  185 CALL BCDEL(I)
      GO TO 220
  190 CALL BCVEL(I)
      GO TO 220
  200 DEL(I)=DELN(I)+(DELN(I)-DELN1(I))*H/HN
      DEL(I)=DEL(I)+XMINV(I)*(BH*B(I)+(H+HN)*H*DIF1*B(I)+BHN*GNM1(I))
      DEL(I)=DEL(I)+XMINV(I)*(BH*F(I)+(H+HN)*H*DIF1*F1(I)+BHN*F2(I))
  220 CONTINUE
  240 DO 480 K=1,NUMNP
C FIND PREDICTED STIFFNESS (EEKM)
  260 IF(NTIME.GT.1) GO TO 300
      IF(K.GT.1) GO TO 300
      I1=IX(1,1)
      I2=IX(1,2)
      I3=IX(1,3)
      I4=IX(1,4)
      AR=SQRT((X(I2)-X(I1))**2+(Y(I2)-Y(I1))**2)
      BR=SQRT((X(I2)-X(I3))**2+(Y(I2)-Y(I3))**2)
      CR=SQRT((X(I4)-X(I3))**2+(Y(I4)-Y(I3))**2)
      FR=SQRT((X(I4)-X(I1))**2+(Y(I4)-Y(I1))**2)
      PR=SQRT((X(I3)-X(I1))**2+(Y(I3)-Y(I1))**2)
```

```
      QR=SQRT((X(I2)-X(I4))**2+(Y(I2)-Y(I4))**2)
      AREA=SQRT(4.0*PR**2*QR**2-(BR**2+FR**2-AR**2-CR**2)**2)/4.0
      REWIND 1
      READ(1)((CRZ(KJ,J),J=1,6),KJ=1,6)
      DO 280 I=1,NUMLA
      EEKM(I)=CRZ(3,3)*(AREA/4.0)/TH(I)
      DO 280 L=1,NUMEL
  280 READ(1)((CRZ(KJ,J),J=1,6),KJ=1,6)
  300 DO 340 I=1,NUMLA
      EKM(I)=EEKM(I)*A(K*3)/A(3)
  340 CONTINUE
C FIND NEW COEFFICIENTS DUE TO STIFFNESS  (KM)
C THIS IS DONE BY SOLVING THE SIMULTANEOUS EQUATIONS THROUGH THE
C THICKNESS BY USE OF A NONSYMMETRIC BANDED MATRIX
      DO 400 I=1,NUP
      II=K*3+(I-1)*NUMNP*3
      ID=CODE(K+(I-1)*NUMNP)
      IF(ID.EQ.0)GO TO 370
      IF(ID.EQ.3.OR.ID.EQ.5.OR.ID.EQ.6.OR.ID.EQ.7)GO TO 350
      IF(ID.EQ.10.OR.ID.EQ.12.OR.ID.EQ.13.OR.ID.EQ.14)GO TO 360
      GO TO 370
  350 CALL BCDEL(II)
      GO TO 380
  360 CALL BCVEL(II)
      GO TO 380
  370 IF(I.EQ.1)GO TO 371
      IF(I.EQ.NUP)GO TO 372
      AU(I,1)=-BH*XMINV(II)*EKM(I-1)
      AU(I,2)=1.0+BH*XMINV(II)*(EKM(I)+EKM(I-1))
      AU(I,3)=-BH*XMINV(II)*EKM(I)
      BU(I,1)=DEL(II)-BH*XMINV(II)*(EKM(I)*DELN(II+NUMNP*3)-(EKM(I)
     1 +EKM(I-1))*DELN(II)+EKM(I-1)*DELN(II-NUMNP*3))
      GO TO 400
  371 AU(I,1)=0.
      AU(I,2)=1.0+BH*XMINV(II)*EKM(I)
      AU(I,3)=-BH*XMINV(II)*EKM(I)
      BU(I,1)=DEL(II)-BH*XMINV(II)*EKM(I)*(DELN(II+NUMNP*3)-DELN(II))
      GO TO 400
  372 AU(I,1)=-BH*XMINV(II)*EKM(I-1)
      AU(I,2)=1.0+BH*XMINV(II)*EKM(I-1)
      AU(I,3)=0.
      BU(I,1)=DEL(II)+BH*XMINV(II)*EKM(I-1)*(DELN(II)-DELN(II-3*NUMNP))
      GO TO 400
  380 AU(I,1)=0.
      AU(I,2)=1.
      AU(I,3)=0.
      BU(I,1)=DEL(II)
  400 CONTINUE
      CALL BAND(AU,BU,NUP,3,2,1,KO)
      IF(KO.NE.0)GO TO 1230
```

```
      DO 420 I=1,NUP
      II=K*3+(I-1)*NUMNP*3
  420 DEL(II)=BU(I,1)
  480 CONTINUE
  500 CONTINUE
      NVD=NUP*2
      DO 1220 K=1,NUMNP
C FIND THE CENTER OF GRAVITY
      IF(NTIME.GT.1) GO TO 780
      IF(K.GT.1) GO TO 780
      AT=A((K-1)*3+1)
      ZCGT=0.0
      THT(1)=0.0
      JL=NUMNP*3+(K-1)*3+1
      DO 520 I=2,NUP
      THT(I)=THT(I-1)+TH(I-1)
      ZCGT=ZCGT+THT(I)*A(JL)
      AT=AT+A(JL)
  520 JL=JL+NUMNP*3
      ZCG=ZCGT/AT
      DO 560 I=1,NUMLA
  560 IF(ZCG.LT.THT(I+1))GO TO 580
  580 NCG=I
      WRITE(6,600)NCG
  600 FORMAT(" CENTER OF GRAVITY IS IN THE ",I5," TH LAYER")
C FIND THE TRANSFORM MATRIX
      DO 620 I=1,NVD
      DO 620 J=1,4
      TF(I,J)=0.0
      IF (I.GT.4) GO TO 620
      TFI(I,J)=0.0
  620 CONTINUE
      DO 660 I=1,NUP
      II=2*(I-1)
      DO 660 J=1,2
      JJ=2*(J-1)+1
      TF(II+J,JJ)=1.0
      KK=(J-1)*2+2
  660 TF(II+J,KK)=(THT(I)-ZCG)
      II=1
      IF (TF(1,2).EQ.TF(3,2)) GO TO  700
      DO  680 I=1,2
      TFI(II,I)=1.0-(TF(1,2)/(TF(1,2)-TF(3,2)))
      J=II+1
      TFI(J,I)=-1.0/(TF(3,2)-TF(1,2))
      III=I+2
      TFI(II,III)=-TF(1,2)/(TF(3,2)-TF(1,2))
      II=II+2
  680 TFI(J,III)=1.0/(TF(3,2)-TF(1,2))
      GO TO  760
```

```
  700 DO  720 I=1,2
      TFI(II,I)=1.0-(TF(5,2)/(TF(5,2)-TF(7,2)))
      J=II+1
      TFI(J,I)=-1.0/(TF(7,2)-TF(5,2))
      III=I+2
      TFI(II,III)=-TF(5,2)/(TF(7,2)-TF(5,2))
      II=II+2
  720 TFI(J,III)=1.0/(TF(5,2)-TF(7,2))
  760 CONTINUE
C TRANSFORM FORCES AND INITIAL DISPLACEMENTS AND VELOCITIES
  780 DO 800 I=1,4
      BOT(I)=0.0
      FOT(I)=0.0
      DELN1T(I)=0.0
      DERT(I)=0.0
      DELNT(I)=0.0
      GNM1T(I)=0.0
  800 GNM2T(I)=0.0
      DO  900 J=1,4
      I=0
      DO  880 IJ=1,NUP
      DO 880 L=1,2
      I=I+1
      NJ=L+3*(K-1)+(IJ-1)*NUMNP*3
      IF (J.GT.1) GO TO 820
      AOT(I)=A(NJ)
  820 BOT(J)=B(NJ)*TF(I,J)+BOT(J)
      FOT(J)=FO(NJ)*TF(I,J)+FOT(J)
      GNM1T(J)=GNM1(NJ)*TF(I,J)+GNM1T(J)
      IF (NTIME.LT.2) GO TO 860
      GNM2T(J)=GNM2(NJ)*TF(I,J)+GNM2T(J)
  860 CONTINUE
  880 CONTINUE
  900 CONTINUE
      DO 960 J=1,4
      I=0
      DO 940 IJ=1,2
      DO 940 L=1,2
      I=I+1
      NJ=(K-1)*3+L+(IJ-1)*NUMNP*3
      DERT(J)=DER(NJ)*TFI(J,I)+DERT(J)
      DELNT(J)=DELN(NJ)*TFI(J,I)+DELNT(J)
      IF (NTIME.LT.2) GO TO  920
      DELN1T(J)=DELN1(NJ)*TFI(J,I)+DELN1T(J)
  920 CONTINUE
  940 CONTINUE
  960 CONTINUE
C FIND NEW MASS MATRIX
      DO  980 I=1,NVD
      DO  980 J=1,4
```

```
  980 AFT(I,J)=AOT(I)*TF(I ,J)
      DO 1000 I=1,4
 1000 SM(I)=0.0
      DO 1120 I=1,4
      DO 1120 J=1,NVD
 1120 SM(I)=SM(I)+TF(J,I)*AFT(J,I)
C FIND DELT
      BH=BET*H**2
      DIF=H**2*(1.0-2.0*BET)
      DIF1=.5-BET
      DO 1180 I=1,4
      IF(NTIME.GT.1) GO TO 1140
      DELT(I)=DELNT(I)+H*DERT(I)+(BH*BOT(I)-DIF1*H**2*GNM1T(I)+DIF1
     1 *H**2*FOT(I))/SM(I)
      GO TO 1180
 1140 DELT(I)=DELNT(I)+(DELNT(I)-DELN1T(I))*H/HN
      DELT(I)=DELT(I)+(BH*BOT(I)+(H+HN)*H*DIF1*GNM1T(I)+BHN*GNM2T(I))
     1 /SM(I)
 1180 CONTINUE
C TRANSFORM DELT TO DEL
      I=0
      DO 1220 IJ=1,NUP
      KJ=K+(IJ-1)*NUMNP
      DO 1220 L=1,2
      II=(K-1)*3+L+(IJ-1)*NUMNP*3
      DEL(II)=0.0
      I=I+1
      ID=CODE(KJ)
      IF(ID.EQ.0)GO TO 1195
      IF((ID.EQ.1.OR.ID.EQ.6).AND.L.EQ.1)GO TO 1185
      IF((ID.EQ.2.OR.ID.EQ.5).AND.L.EQ.2)GO TO 1185
      IF(ID.EQ.4.OR.ID.EQ.7)GO TO 1185
      IF((ID.EQ.8.OR.ID.EQ.13).AND.L.EQ.1)GO TO 1190
      IF((ID.EQ.9.OR.ID.EQ.12).AND.L.EQ.2)GO TO 1190
      IF(ID.EQ.11.OR.ID.EQ.14)GO TO 1190
      GO TO 1195
 1185 CALL BCDEL(II)
      GO TO 1220
 1190 CALL BCVEL(II)
      GO TO 1220
 1195 DO 1200 J=1,4
 1200 DEL(II)=DEL(II)+TF(I,J)*DELT(J)
 1220 CONTINUE
      RETURN
 1230 WRITE(6,1250)
 1250 FORMAT(" * * * * * ERROR IN SUBROUTINE BAND * * * * *"/
     1 " * * * * * CHECK KO OR BOUNDARY CONDITIONS * * * * *")
      END
      SUBROUTINE INTER
      INTEGER CODE
```

```
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA,NCG
      COMMON/MATP/RO(12),E(9,12),EE(9),ETA(12)
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
      COMMON/RESULT/D(6,6),C(6,6),CNS(6,6)
      COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
      COMMON/BASIC2/BET,H,HN,HH(3),HT(2),TIME,NLAY,IFLAG,IT,NTIME
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
      COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
      COMMON/MASS/A(600),B(600),CM(8)
      DIMENSION X(7),Y(7),QQ(9)
      DATA QQ/3*.1259391805448,3*.1323941527884,.225,
     1 .696140478028,.410426192314/
      X(7)=(XX(1)+XX(2)+XX(3))/3.0
      Y(7)=(YY(1)+YY(2)+YY(3))/3.0
      DO 100 I=1,3
      J=I+3
      X(I)=QQ(8)*XX(I)+(1.00-QQ(8))*X(7)
      X(J)=QQ(9)*XX(I)+(1.00-QQ(9))*X(7)
      Y(I)=QQ(8)*YY(I)+(1.00-QQ(8))*Y(7)
  100 Y(J)=QQ(9)*YY(I)+(1.00-QQ(9))*Y(7)
      DO 300 I=1,10
  300 XI(I)=0.00
      AREA=.50*(XX(1)*(YY(2)-YY(3))+XX(2)*(YY(3)-YY(1))+XX(3)*(YY(1)
     1  -YY(2)))
      DO 400 I=1,7
      XI(1)=XI(1)+QQ(I)
      XI(2)=XI(2)+QQ(I)*X(I)
      XI(3)=XI(3)+QQ(I)*Y(I)
      XI(4)=XI(4)+QQ(I)*X(I)*Y(I)
      XI(5)=XI(5)+QQ(I)*X(I)**2
  400 XI(6)=XI(6)+QQ(I)*Y(I)**2
      DO 500 I=1,10
  500 XI(I)=XI(I)*AREA
      RETURN
      END
      SUBROUTINE LOAD
      INTEGER CODE
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA,NCG
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/NPDATA/X(200),Y(200),CODE(200),NPNUM(10,20)
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
      COMMON/BASIC2/BET,H,HN,HH(3),HT(2),TIME,NLAY,IFLAG,IT,NTIME
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
      COMMON/FOR/FF(5),FC(5)
      DO 50 I=1,IT
      F(I)=0.00
 50   CONTINUE
```

```
C     NUMBER NODES OF THE RECTANGULAR ELEMENTS
      DO 20 N=1,NUMEL
      I1=IX(N,1)
      I2=IX(N,2)
      I3=IX(N,3)
      I4=IX(N,4)
C     DESIGNATE COORDINATES
      XXX(1)=X(I1)
      XXX(2)=X(I2)
      XXX(3)=X(I3)
      XXX(4)=X(I4)
      XXX(5)=1.0/4.0*(XXX(1)+XXX(2)+XXX(3)+XXX(4))
      YYY(1)=Y(I1)
      YYY(2)=Y(I2)
      YYY(3)=Y(I3)
      YYY(4)=Y(I4)
      YYY(5)=1.0/4.0*(YYY(1)+YYY(2)+YYY(3)+YYY(4))
      CALL DISFOR
      IF(FF(1).EQ.0.0.AND.FF(2).EQ.0.0.AND.FF(3).EQ.0.0.AND.FF(4).EQ.0.0
     1) GO TO 20
C     CALL LOADS OF TRIANGLES
      DO 9 I=1,5
      FC(I)=0.00
    9 CONTINUE
      CALL LOT(1,2)
      CALL LOT(2,3)
      CALL LOT(3,4)
      CALL LOT(4,1)
      DO 21 I=1,4
      FC(I)=FC(I)+FC(5)/4.0
   21 CONTINUE
C     CHANGE TO GLOBAL FORCES IN W-DIRECTION
      II1=NUMLA*NUMNP*3+I1*3
      II2=NUMLA*NUMNP*3+I2*3
      II3=NUMLA*NUMNP*3+I3*3
      II4=NUMLA*NUMNP*3+I4*3
      F(II1)=F(II1)+FC(1)
      F(II2)=F(II2)+FC(2)
      F(II3)=F(II3)+FC(3)
      F(II4)=F(II4)+FC(4)
   20 CONTINUE
      RETURN
      END
      SUBROUTINE LOT(II,JJ)
      INTEGER CODE
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA,NCG
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/NPDATA/X(200),Y(200),CODE(200),NPNUM(10,20)
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
```

```
      COMMON/BASIC2/BET,H,HN,HH(3),HT(2),TIME,NLAY,IFLAG,IT,NTIME
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
      COMMON/FOR/FF(5),FC(5)
      DIMENSION DD1(3),DD2(3),DD3(3),DD(3,3),FE(3),FFF(3)
      DIMENSION AA(3),BB(3),CC(3)
C     DESIGNATE THE TRIANGULAR DISTRIBUTED FORCES
      FFF(1)=FF(II)
      FFF(2)=FF(JJ)
      FFF(3)=FF(5)
      XX(1)=XXX(II)
      XX(2)=XXX(JJ)
      XX(3)=XXX(5)
      YY(1)=YYY(II)
      YY(2)=YYY(JJ)
      YY(3)=YYY(5)
      AA(1)=XX(2)*YY(3)-XX(3)*YY(2)
      AA(2)=XX(3)*YY(1)-XX(1)*YY(3)
      AA(3)=XX(1)*YY(2)-XX(2)*YY(1)
      BB(1)=YY(2)-YY(3)
      BB(2)=YY(3)-YY(1)
      BB(3)=YY(1)-YY(2)
      CC(3)=XX(2)-XX(1)
      CC(2)=XX(1)-XX(3)
      CC(1)=XX(3)-XX(2)
C     INTEGRATE XX AND YY
      CALL INTER
      DO 12 I=1,3
      DD1(I)=AA(I)*XI(1)+BB(I)*XI(2)+CC(I)*XI(3)
      DD2(I)=AA(I)*XI(2)+BB(I)*XI(5)+CC(I)*XI(4)
      DD3(I)=AA(I)*XI(3)+BB(I)*XI(4)+CC(I)*XI(6)
   12 CONTINUE
      DO 18 I=1,3
      DO 18 J=1,3
      DD(I,J)=AA(I)*DD1(J)+BB(I)*DD2(J)+CC(I)*DD3(J)
   18 CONTINUE
C     CALCULATE EQUIVELENT CONCENTRATED FORCES
      AREA=.50*(XX(1)*(YY(2)-YY(3))+XX(2)*(YY(3)-YY(1))+XX(3)*(YY(1)
     1  -YY(2)))
      DO 99 I=1,3
      FE(I)=1.0/(4.0*AREA**2)*(DD(I,1)*FFF(1)+DD(I,2)*FFF(2)+DD(I,3)*FFF
     1(3))
   99 CONTINUE
      FC(II)=FC(II)+FE(1)
      FC(JJ)=FC(JJ)+FE(2)
      FC(5)=FC(5)+FE(3)
      RETURN
      END
      SUBROUTINE MESH
      INTEGER CODE
      DIMENSION AR(10, 40),AZ(10, 40),NCODE(10, 40)
```

```
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA,NCG
      COMMON/MATP/RO(12),E(9,12),EE(9),ETA(12)
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/NPDATA/X(200),Y(200),CODE(200),NPNUM(10,20)
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
      COMMON/RESULT/D(6,6),C(6,6),CNS(6,6)
      COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
      COMMON/BASIC2/BET,H,HN,HH(3),HT(2),TIME,NLAY,IFLAG,IT,NTIME
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
      COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
      COMMON/MASS/A(600),B(600),CM(8)
      EQUIVALENCE (X(1),AR),(Y(1),AZ),(IX(1,1),NCODE)
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     MESH CONTROL INFORMATION
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      READ (5,1000) MAXI,MAXJ,NSEG,NBC,NMTL
      WRITE(6,2000) MAXI,MAXJ,NSEG,NBC,NMTL
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     INITIALIZE
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      ISEG=-1
      PI=3.1415927
      DO 110 J=1,40
      DO 100 I=1,10
      NCODE(I,J)=0
      AR(I,J)=0.
      AZ(I,J)=0.
      JMAX(I)=0
  100 JMIN(I)=MAXI
      IMIN(J)=MAXJ
  110 IMAX(J)=0
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     LINE SEGMENT CARDS
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
  150 ISEG=ISEG+1
  159 IF(ISEG.EQ.NSEG) GO TO 400
      READ(5,1001) I1,J1,R1,Z1,I2,J2,R2,Z2,I3,J3,R3,Z3,IPTION
      WRITE(6,2001)I1,J1,R1,Z1,I2,J2,R2,Z2,I3,J3,R3,Z3,IPTION
      IPTION=IPTION+1
      AR(I1,J1)=R1
      AZ(I1,J1)=Z1
      NCODE(I1,J1)=1
      CALL MNIMX(I1,J1)
      GO TO (150,200,200,300,300,200,200), IPTION
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     GENERATE STRAIGHT LINES ON BOUNDARY
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
200   DI= ABS(FLOAT(I2-I1))
      DJ= ABS(FLOAT(J2-J1))
```

```
      AR(I2,J2)=R2
      AZ(I2,J2)=Z2
      NCODE(I2,J2)=1
      CALL MNIMX(I2,J2)
      ISTRT=I1
      ISTP=I2
      JSTRT=J1
      JSTP=J2
      DIFF=MAX1(DI,DJ)
      ITER=DIFF-1.
      IINC=0
      JINC=0
      IF(I2.NE.I1) IINC=(I2-I1)/IABS(I2-I1)
      IF(J2.NE.J1) JINC=(J2-J1)/IABS(J2-J1)
      KAPPA=1
      IF(I2.NE.I1.AND.J2.NE.J1.AND.IPTION.NE.3) KAPPA=2
      IF(KAPPA.EQ.2) DIFF=2.*DIFF
      RINC=(R2-R1)/DIFF
      ZINC=(Z2-Z1)/DIFF
      WRITE(6,2002) DI,DJ,DIFF,RINC,ZINC,ITER,IINC,JINC,KAPPA
C
C     CHECK FOR INPUT ERROR
C
      IF(KAPPA.NE.2.OR.DI.EQ.DJ) GO TO 210
      WRITE(6,2003)
      GO TO 150
C
C     INTERPOLATE
C
  210 I=I1
      J=J1
      WRITE(6,2004)
      DO 230 M=1,ITER
      IF(ITER.EQ.0.AND.IPTION.EQ.2) GO TO 230
      IF(ITER.EQ.0.AND.IPTION.EQ.6) GO TO 230
      IF(ITER.EQ.0.AND.IPTION.EQ.7) GO TO 230
      IF(KAPPA.EQ.2) GO TO 220
      IOLD=I
      I=I+IINC
      JOLD=J
      J=J+JINC
      AR(I,J)=AR(IOLD,JOLD)+RINC
      AZ(I,J)=AZ(IOLD,JOLD)+ZINC
      WRITE(6,2005) I,J,AR(I,J),AZ(I,J)
      CALL MNIMX(I,J)
      NCODE(I,J)=1
      GO TO 230
  220 CONTINUE
      IF(I1.GT.I2.AND.IPTION.EQ.7) GO TO 221
      IF(I1.LT.I2.AND.IPTION.EQ.6) GO TO 221
```

```
      IOLD=I
      I=I+IINC
      AR(I,J)=AR(IOLD,J)+RINC
      AZ(I,J)=AZ(IOLD,J)+ZINC
      WRITE(6,2005) I,J,AR(I,J),AZ(I,J)
      NCODE(I,J)=1
      CALL MNIMX(I,J)
      JOLD=J
      J=J+JINC
      AR(I,J)=AR(I,JOLD)+RINC
      AZ(I,J)=AZ(I,JOLD)+ZINC
      NCODE(I,J)=1
      WRITE(6,2005) I,J,AR(I,J),AZ(I,J)
      CALL MNIMX(I,J)
      GO TO 230
  221 JOLD=J
      J=J+JINC
      AR(I,J)=AR(I,JOLD)+RINC
      AZ(I,J)=AZ(I,JOLD)+ZINC
      NCODE(I,J)=1
      WRITE(6,2005) I,J,AR(I,J),AZ(I,J)
      CALL MNIMX(I,J)
      IOLD=I
      I=I+IINC
      AR(I,J)=AR(IOLD,J)+RINC
      AZ(I,J)=AZ(IOLD,J)+ZINC
      NCODE(I,J)=1
      WRITE(6,2005) I,J,AR(I,J),AZ(I,J)
      CALL MNIMX(I,J)
  230 CONTINUE
      IF(KAPPA.EQ.1) GO TO 150
      IF(I1.GT.I2.AND.IPTION.EQ.7) GO TO 231
      IF(I1.LT.I2.AND.IPTION.EQ.6) GO TO 231
        IOLD=I
      I=I+IINC
      AR(I,J)=AR(IOLD,J)+RINC
      AZ(I,J)=AZ(IOLD,J)+ZINC
      GO TO 232
  231 CONTINUE
      JOLD=J
      J=J+JINC
      AR(I,J)=AR(I,JOLD)+RINC
      AZ(I,J)=AZ(I,JOLD)+ZINC
  232 CONTINUE
      NCODE(I,J)=1
      WRITE(6,2005) I,J,AR(I,J),AZ(I,J)
      CALL MNIMX(I,J)
      GO TO 150
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     GENERATE CIRCULAR ARCS ON BOUNDARY
```

```
C
      DI= ABS(FLOAT(ISTP-ISTRT))
      DJ= ABS(FLOAT(JSTP-JSTRT))
      IINC=0
      JINC=0
      IF(ISTRT.NE.ISTP) IINC=(ISTP-ISTRT)/IABS(ISTP-ISTRT)
      IF(JSTRT.NE.JSTP) JINC=(JSTP-JSTRT)/IABS(JSTP-JSTRT)
      LAMDA=1
      IF(IINC.NE.0.AND.JINC.NE.0) LAMDA=2
      DIFF=MAX1(DI,DJ)
      ITER=DIFF-1.
      IF(LAMDA.EQ.2) DIFF=2.*DIFF
      DELPHI=(ANG2-ANG1)/DIFF
      WRITE(6,2008) ANG1,ANG2,DIFF,DELPHI
C
C     CHECK FOR INPUT ERROR
C
      IF(LAMDA.NE.2.OR.DI.EQ.DJ) GO TO 350
      WRITE(6,2003)
      GO TO 150
  350 IO=ISTRT
      JO=JSTRT
      WRITE(6,2004)
C
C     INTERPOLATE
C
      NPT=IABS(I2-I1)+IABS(J2-J1)-1
      DO 380 M=1,ITER
  359 IF(LAMDA.EQ.2) GO TO 360
      I=IO+IINC
      J=JO+JINC
      CALL MNIMX(I,J)
      NCODE(I,J)=1
      CALL CIRCLE(ANG1,DELPHI,RSTRT,ZSTRT,RC,ZC,I,J)
      WRITE(6,2005) I,J,AR(I,J),AZ(I,J)
      GO TO 370
  360 I=IO+IINC
      J=JO
      NCODE(I,J)=1
      CALL MNIMX(I,J)
      CALL CIRCLE(ANG1,DELPHI,RSTRT,ZSTRT,RC,ZC,I,J)
      WRITE(6,2005) I,J,AR(I,J),AZ(I,J)
      J=JO+JINC
      NCODE(I,J)=1
      CALL MNIMX(I,J)
      CALL CIRCLE(ANG1,DELPHI,RSTRT,ZSTRT,RC,ZC,I,J)
      WRITE(6,2005) I,J,AR(I,J),AZ(I,J)
  370 IO=I
  380 JO=J
      IF(LAMDA.NE.2) GO TO 390
```

```
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
  300 AR(I2,J2)=R2
      AZ(I2,J2)=Z2
      NCODE(I2,J2) = 1
      CALL MNIMX(I2,J2)
      IF(IPTION.EQ.5) GO TO 320
C
C     FIND CENTER OF CIRCLE
C
      AR(I3,J3)=R3
      AZ(I3,J3)=Z3
      NCODE(I3,J3)=1
      CALL MNIMX(I3,J3)
      SLAC=(Z2-Z1)/(R2-R1)
      SLBF=-1./SLAC
      SLCE=(Z3-Z2)/(R3-R2)
      SLDF=-1./SLCE
C
C     CHECK FOR INPUT ERROR
C
      IF(ABS(SLAC-SLCE).GT..001) GO TO 310
      WRITE(6,2006) R1,Z1,R2,Z2,R3,Z3,SLAC,SLCE
      GO TO 150
  310 R4=R1+(R2-R1)/2.
      Z4=Z1+(Z2-Z1)/2.
      R5=R2+(R3-R2)/2.
      Z5=Z2+(Z3-Z2)/2.
      BBF=Z4-SLBF*R4
      BDF=Z5-SLDF*R5
      RC=(BBF-BDF)/(SLDF-SLBF)
      ZC=SLBF*RC+BBF
      WRITE(6,2007) RC,ZC
      KAPPA=1
      GO TO 330
  320 KAPPA=2
      RC=R3
      ZC=Z3
  330 ISTRT=I1
      ISTP=I2
      JSTRT=J1
      JSTP=J2
      RSTRT=R1
      RSTP=R2
      ZSTRT=Z1
      ZSTP=Z2
  340 CALL ANGLE(RSTRT,ZSTRT,RC,ZC,ANG1)
      CALL ANGLE(RSTP,ZSTP,RC,ZC,ANG2)
      IF(ANG2.LE.ANG1)  ANG2=2.0*PI+ANG2
C
C     FIND ANGULAR INCREMENT
```

```
      I=IO+IINC
      NCODE(I,J)=1
      CALL MNIMX(I,J)
      CALL CIRCLE(ANG1,DELPHI,RSTRT,ZSTRT,RC,ZC,I,J)
      WRITE(6,2005) I,J,AR(I,J),AZ(I,J)
  390 IF(KAPPA.EQ.2) GO TO 150
      ISTRT=I2
      ISTP=I3
      JSTRT=J2
      JSTP=J3
      RSTRT=R2
      RSTP=R3
      ZSTRT=Z2
      ZSTP=Z3
      KAPPA=2
  399 GO TO 340
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     CALCULATE COORDINATES OF INTERIOR POINTS
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
  400 IF(MAXJ.LE.2) GO TO 430
      J2=MAXJ-1
      DO 420 N=1,500
      RESID=0.
      DO 410 J=2,J2
      I1=IMIN(J)+1
      I2=IMAX(J)-1
      DO 410 I=I1,I2
      IF(NCODE(I,J).EQ.1) GO TO 410
      DR=(AR(I+1,J)+AR(I-1,J)+AR(I,J+1)+AR(I,J-1))/4.-AR(I,J)
      DZ=(AZ(I+1,J)+AZ(I-1,J)+AZ(I,J+1)+AZ(I,J-1))/4.-AZ(I,J)
      RESID=RESID+ABS(DR)+ABS(DZ)
      AR(I,J)=AR(I,J)+1.8*DR
      AZ(I,J)=AZ(I,J)+1.8*DZ
  410 CONTINUE
      IF(N.EQ.1) RES1=RESID
      IF(N.EQ.1.AND.RESID.EQ.0.)GO TO 430
      IF(RESID/RES1.LT.1.E-5) GO TO 430
  420 CONTINUE
  430 WRITE(6,2009) N
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      CALL POINTS
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
 1000 FORMAT (5I5)
 1001 FORMAT (3(2I3,2F8.3),I5)
 2000 FORMAT (30H1  MESH GENERATION INFORMATION//
     1 41H0  MAXIMUM VALUE OF I IN THE MESH--------I3/
     2 41H0  MAXIMUM VALUE OF J IN THE MESH--------I3/
     3 41H0  NUMBER OF LINE SEGMENT CARDS----------I3/
     4 41H0  NUMBER OF BOUNDARY CONDITION CARDS----I3/
     5 41H0  NUMBER OF MATERIAL BLOCK CARDS--------I3///)
```

```
2001 FORMAT (//88H  INPUT   I1  J1    R1      Z1    I2  J2    R2      Z
    12    I3  J3    R3      Z3    IPTION/8X,3(2I4,2F8.4),I6)
2002 FORMAT (5H  DI=F4.0,5H  DJ=F4.0,7H  DIFF=F4.0,7H  RINC=F8.3,7H  ZI
    1NC=F8.3,7H  ITER=I3,7H  IINC=I3,7H  JINC=I3,8H  KAPPA=I1)
2003 FORMAT(1X,38H**BAD INPUT--THIS LINE IS NOT DIAGONAL)
2004 FORMAT (30H    I    J        AR        AZ)
2005 FORMAT (2I5,2F11.6)
2006 FORMAT (51H ** BAD INPUT - THESE POINTS DO NOT DEFINE A CIRCLE,/,
    13X,6F12.4,10X,2E20.8)
2007 FORMAT(19H  CENTER COORDINATE,(F11.6,1X,F11.6,1X))
2008 FORMAT (7H  ANG1=F9.6,7H  ANG2=F9.6,7H  DIFF=F3.0,9H  DELPHI=F9.6)
2009 FORMAT (//30H  COORDINATES CALCULATED AFTER I3,11H ITERATIONS)
      RETURN
      END
      SUBROUTINE MNIMX(I,J)
      INTEGER CODE
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA,NCG
      COMMON/MATP/RO(12),E(9,12),EE(9),ETA(12)
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/NPDATA/X(200),Y(200),CODE(200),NPNUM(10,20)
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
      COMMON/RESULT/D(6,6),C(6,6),CNS(6,6)
      COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
      COMMON/BASIC2/BET,H,HN,HH(3),HT(2),TIME,NLAY,IFLAG,IT,NTIME
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
      COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
      COMMON/MASS/A(600),B(600),CM(8)
      IF(J.LT.JMIN(I)) JMIN(I)=J
      IF(J.GT.JMAX(I)) JMAX(I)=J
      IF(I.LT.IMIN(J)) IMIN(J)=I
      IF(I.GT.IMAX(J)) IMAX(J)=I
      RETURN
      END
      SUBROUTINE MPLOT
      INTEGER CODE
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA,NCG
      COMMON/MATP/RO(12),E(9,12),EE(9),ETA(12)
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/NPDATA/R(200),Z(200),CODE(200),NPNUM(10,20)
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
      COMMON/RESULT/D(6,6),C(6,6),CNS(6,6)
      COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
      COMMON/BASIC2/BET,H,HN,HH(3),HT(2),TIME,NLAY,IFLAG,IT,NTIME
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
      COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
      COMMON/MASS/A(600),B(600),CM(8)
      DIMENSION TITLE(8),THI(13)
      IF(NTIME.GT.0)GO TO 50
```

```
      READ(5,100)TITLE,RMAX,ZMAX
C     WRITE(6,100)TITLE,RMAX,ZMAX
      TITLE(8)=";"
   50 CONTINUE
  100 FORMAT(8A10/2F10.5)
      NUP=NUMLA+1
      S3=.5
      C3=SQRT(3.)/2.
      THI(1)=0.0
      DO 200 I=1,NUMLA
  200 THI(I+1)=TH(I)+THI(I)
      XMAX=C3*RMAX*1.1
      XMIN=-C3*ZMAX*1.1
      YMAX=(XMAX-XMIN)*3./4.
      YMIN=-YMAX/3.
C DRAW X LINES
      CALL UWINDO(XMIN,XMAX,YMIN,YMAX)
      YMIN=YMIN/4.
      CALL UPRINT(XMIN,0.,TITLE)
      CALL UPRINT(XMIN,YMIN,"  TIME = ;")
      CALL UPRNT1(TIME,"REAL")
      DO 300 J=1,MAXJ
      NSTART=IMIN(J)
      NSTOP=IMAX(J)
      DO 300 K=1,NUP
      IF(J.GT.1.AND.K.NE.NUP)GO TO 300
      DO 250 I=NSTART,NSTOP
      NP=NPNUM(I,J)
      X=C3*(R(NP)-Z(NP))
      Y=THI(K)+S3*(R(NP)+Z(NP))
      IF(I.EQ.NSTART)CALL UMOVE(X,Y)
      CALL UPEN(X,Y)
  250 CONTINUE
  300 CONTINUE
C DRAW Y LINES
      DO 400 I=1,MAXI
      NSTART=JMIN(I)
      NSTOP=JMAX(I)
      DO 400 K=1,NUP
      IF(I.GT.1.AND.K.NE.NUP)GO TO 400
      DO 350 J=NSTART,NSTOP
      NP=NPNUM(I,J)
      X=C3*(R(NP)-Z(NP))
      Y=THI(K)+S3*(R(NP)+Z(NP))
      IF(J.EQ.NSTART)CALL UMOVE(X,Y)
      CALL UPEN(X,Y)
  350 CONTINUE
  400 CONTINUE
C DRAW Z LINES
      DO 500 I=1,MAXI
```

```
      NP=NPNUM(I,1)
      DO 500 K=1,NUP
      X=C3*(R(NP)-Z(NP))
      Y=S3*(R(NP)+Z(NP))+THI(K)
      IF(K.EQ.1)CALL UMOVE(X,Y)
  500 CALL UPEN(X,Y)
      DO 600 J=1,MAXJ
      NP=NPNUM(1,J)
      DO 600 K=1,NUP
      X=C3*(R(NP)-Z(NP))
      Y=S3*(R(NP)+Z(NP))+THI(K)
      IF(K.EQ.1)CALL UMOVE(X,Y)
  600 CALL UPEN(X,Y)
      IF(NTIME.EQ.0)GO TO 650
      CALL MPLOTD(ZMAX,RMAX)
      GO TO 800
C LABEL NODES
  650 CALL USET("INTEGER")
      DO 700 I=1,MAXI
      DO 700 J=1,MAXJ
      DO 700 K=1,NUP
      NP=NPNUM(I,J)
      X=C3*(R(NP)-Z(NP))
      Y=S3*(R(NP)+Z(NP))+THI(K)
      P1=NP+(K-1)*NUMNP
  700 IF(I.EQ.1.OR.J.EQ.1.OR.K.EQ.NUP)CALL UPRINT(X,Y,P1)
      CALL USET("TEXT")
  800 CALL UERASE
      RETURN
      END
      SUBROUTINE MPLOTD(ZMAX,RMAX)
      INTEGER CODE
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA,NCG
      COMMON/MATP/RO(12),E(9,12),EE(9),ETA(12)
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/NPDATA/R(200),Z(200),CODE(200),NPNUM(10,20)
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
      COMMON/RESULT/D(6,6),C(6,6),CNS(6,6)
      COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
      COMMON/BASIC2/BET,H,HN,HH(3),HT(2),TIME,NLAY,IFLAG,IT,NTIME
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
      COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
      COMMON/MASS/A(600),B(600),CM(8)
      DIMENSION TITLE(8),THI(13)
      CALL USET("DASH")
      NUP=NUMLA+1
      S3=.5
      C3=SQRT(3.)/2.
      THI(1)=0.0
```

```
      DO 200 I=1,NUMLA
  200 THI(I+1)=TH(I)+THI(I)
C DRAW X LINES
      DO 300 J=1,MAXJ
      NSTART=IMIN(J)
      NSTOP=IMAX(J)
      DO 300 K=1,NUP
      IF(J.GT.1.AND.K.NE.NUP)GO TO 300
      DO 250 I=NSTART,NSTOP
      NP=NPNUM(I,J)
      NP1=(NP+(K-1)*NUMNP)*3
      X=C3*(R(NP)+DEL(NP1-2)-Z(NP)+DEL(NP1-1))
      Y=THI(K)+DEL(NP1)+S3*(R(NP)+DEL(NP1-2)+Z(NP)+DEL(NP1-1))
      IF(I.EQ.NSTART)CALL UMOVE(X,Y)
      CALL UPEN(X,Y)
  250 CONTINUE
  300 CONTINUE
C DRAW Y LINES
      DO 400 I=1,MAXI
      NSTART=JMIN(I)
      NSTOP=JMAX(I)
      DO 400 K=1,NUP
      IF(I.GT.1.AND.K.NE.NUP)GO TO 400
      DO 350 J=NSTART,NSTOP
      NP=NPNUM(I,J)
      NP1=(NP+(K-1)*NUMNP)*3
      X=C3*(R(NP)+DEL(NP1-2)-Z(NP)+DEL(NP1-1))
      Y=THI(K)+DEL(NP1)+S3*(R(NP)+DEL(NP1-2)+Z(NP)+DEL(NP1-1))
      IF(J.EQ.NSTART)CALL UMOVE(X,Y)
      CALL UPEN(X,Y)
  350 CONTINUE
  400 CONTINUE
C DRAW Z LINES
      DO 500 I=1,MAXI
      NP=NPNUM(I,1)
      DO 500 K=1,NUP
      NP1=(NP+(K-1)*NUMNP)*3
      X=C3*(R(NP)+DEL(NP1-2)-Z(NP)+DEL(NP1-1))
      Y=S3*(R(NP)+DEL(NP1-2)+Z(NP)+DEL(NP1-1))+THI(K)+DEL(NP1)
      IF(K.EQ.1)CALL UMOVE(X,Y)
  500 CALL UPEN(X,Y)
      DO 600 J=1,MAXJ
      NP=NPNUM(1,J)
      DO 600 K=1,NUP
      NP1=(NP+(K-1)*NUMNP)*3
      X=C3*(R(NP)+DEL(NP1-2)-Z(NP)+DEL(NP1-1))
      Y=S3*(R(NP)+DEL(NP1-2)+Z(NP)+DEL(NP1-1))+THI(K)+DEL(NP1)
      IF(K.EQ.1)CALL UMOVE(X,Y)
  600 CALL UPEN(X,Y)
      CALL USET("LINE")
```

```
      RETURN
      END
      FUNCTION NODE(I,J)
      INTEGER CODE
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA,NCG
      COMMON/MATP/RO(12),E(9,12),EE(9),ETA(12)
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/NPDATA/X(200),Y(200),CODE(200),NPNUM(10,20)
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
      COMMON/RESULT/D(6,6),C(6,6),CNS(6,6)
      COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
      COMMON/BASIC2/BET,H,HN,HH(3),HT(2),TIME,NLAY,IFLAG,IT,NTIME
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
      COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
      COMMON/MASS/A(600),B(600),CM(8)
      NODE=0
      DO 100 JJ=1,J
      NSTART=IMIN(JJ)
      NSTOP=IMAX(JJ)
      DO 100 II=NSTART,NSTOP
      NODE=NODE+1
      IF(JJ.EQ.J.AND.II.EQ.I) RETURN
  100 CONTINUE
      RETURN
      END
      SUBROUTINE POINT(TIME,NUM,TN)
      COMMON/NUMB/NITER,HDAT,TO
      AN=(TIME-TO)/HDAT+1.0
      NUM=AN
      NUM=(AN+NUM+1.5)/2
      TN=TO+(NUM-1)*HDAT
      RETURN
      END
      SUBROUTINE POINTS
      INTEGER CODE
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA,NCG
      COMMON/MATP/RO(12),E(9,12),EE(9),ETA(12)
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/NPDATA/X(200),Y(200),CODE(200),NPNUM(10,20)
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
      COMMON/RESULT/D(6,6),C(6,6),CNS(6,6)
      COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
      COMMON/BASIC2/BET,H,HN,HH(3),HT(2),TIME,NLAY,IFLAG,IT,NTIME
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
      COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
      COMMON/MASS/A(600),B(600),CM(8)
      COMMON/PLYLD/SIGY(12,6),DEPS(6),EPS(12,50,6)
      DIMENSION AX(10,20),AY(10,20),BLKANG(12),BLKALF(12)
```

```
      EQUIVALENCE (X(1),AX),(Y(1),AY)
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     ESTABLISH NODAL POINT INFORMATION
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      NEL=0
      NODSUM=0
      DO 100 J=1,MAXJ
      NSTART=IMIN(J)
      NSTOP=IMAX(J)
      DO 100 I=NSTART,NSTOP
  100 NODSUM=NODSUM+1
      NELSUM=0
      JJMAX=MAXJ-1
      DO 110 JJ=1,JJMAX
      NSTOP=MIN0(IMAX(JJ),IMAX(JJ+1))-1
      NSTART=MAX0(IMIN(JJ),IMIN(JJ+1))
      DO 110 II=NSTART,NSTOP
  110 NELSUM=NELSUM+1
      NUMNP=NODSUM
      NUMEL=NELSUM
      DO 120 J=1,MAXJ
      NSTART=IMIN(J)
      NSTOP=IMAX(J)
      DO 120 I=NSTART,NSTOP
      NPNUM(I,J)=NODE(I,J)
      NP=NPNUM(I,J)
      X(NP)=AX(I,J)
  120 Y(NP)=AY(I,J)
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     READ AND ASSIGN BOUNDARY CONDITIONS
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     INITIALIZE
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      NUMN=(NUMLA+1)*NUMNP
      DO 130 I=1,NUMN
      CODE(I)=0
  130 CONTINUE
      IF(NBC.EQ.0) GO TO 210
      DO 200 IBCON=1,NBC
      READ(5,1002)N1,N2,ICN
      DO 200 I=N1,N2
      CODE(I)=ICN
  200 CONTINUE
  210 MPRINT=0
      DO 230 J=1,MAXJ
      NSTART=IMIN(J)
      NSTOP=IMAX(J)
      DO 230 I=NSTART,NSTOP
      NP=NPNUM(I,J)
      IF(MPRINT.NE.0) GO TO 220
```

```
      WRITE(6,2000)
      MPRINT=59
  220 MPRINT=MPRINT-1
  230 WRITE(6,2001) I,J,NP,CODE(NP),X(NP),Y(NP)
      DO 310 IMTL=1,NUMLA
      READ (5,1000) MTL,BETA(IMTL),ALPHA(IMTL),ETA(IMTL)
      READ(5,2444)(SIGY(IMTL,I),I=1,6)
 2444 FORMAT(6E12.6)
      IF(SIGY(IMTL,2).NE.0.0) GO TO 244
      DO 242 I=2,3
  242 SIGY(IMTL,I)=SIGY(IMTL,1)
      DO 243 I=4,6
  243 SIGY(IMTL,I)=SIGY(IMTL,1)/SQRT(3.0)
  244 CONTINUE
  310 MATRIL(IMTL)=MTL
      WRITE(6,2003) (CODE(I),I=1,NUMN)
 2003 FORMAT(25I3)
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     ESTABLISH ELEMENT INFORMATION
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      JJMAX=MAXJ-1
      N=0
      MTL=1
      KTL=1
      DO 440 JJ=1,JJMAX
      NSTOP=MIN0(IMAX(JJ),IMAX(JJ+1))-1
      NSTART=MAX0(IMIN(JJ),IMIN(JJ+1))
      DO 440 II=NSTART,NSTOP
      NEL=NEL+1
  420 I=NPNUM(II,JJ)
      J=I+1
      K=NPNUM(II+1,JJ+1)
      L=K-1
      M=NEL
      IX(M,1)=I
      IX(M,2)=J
      IX(M,3)=K
      IX(M,4)=L
  440 CONTINUE
      IF(NUMNP.GT.2000) WRITE(6,2002)
 1000 FORMAT ( I5,2F10.0,E12.6)
 1002 FORMAT(2I5,I10)
 2000 FORMAT (59H1    I    J    NP        TYPE    X-ORDINATE    Z-ORDINA
     1TE  )
 2001 FORMAT (2I5,I6,I12,F13.6,F14.6,E26.7,E24.7,E24.7)
 2002 FORMAT (35H  BAD INPUT - TOO MANY NODAL POINTS)
      RETURN
      END
      SUBROUTINE QUAD
      INTEGER CODE
```

```
      REAL NUSN,NUTN,NUTS,NUNS,NUNT,NUST
      DIMENSION DUMMY(6,6),DUMMY1(6,6)
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA,NCG
      COMMON/MATP/RO(12),E(9,12),EE(9),ETA(12)
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/NPDATA/X(200),Y(200),CODE(200),NPNUM(10,20)
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
      COMMON/RESULT/D(6,6),C(6,6),CNS(6,6)
      COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
      COMMON/BASIC2/BET,H,HN,HH(3),HT(2),TIME,NLAY,IFLAG,IT,NTIME
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
      COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
      COMMON/MASS/A(600),B(600),CM(8)
       IF(NTIME.EQ.1) GO TO 10
      READ(1) ((CRZ(I,J),J=1,6),I=1,6)
       GO TO 151
   10  CONTINUE
      I1=IX(N,1)
      J1=IX(N,2)
      K1=IX(N,3)
      L1=IX(N,4)
      MTYPE=MATRIL(NLAY)
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     INTERPOLATE MATERIAL PROPERTIES
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      DO 100 I=1,9
  100 EE(I)=E(I,MTYPE)
      DO 110 I=1,6
      DO 110 J=1,6
      CNS(I,J)=0.00
      C(I,J)=0.00
  110 D(I,J)=0.00
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C     FORM STRESS-STRAIN RELATIONSHIP IN N-S-T SYSTEM
C* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
      NUNS=EE(4)
      NUNT=EE(5)
      NUST=EE(6)
      NUSN=(EE(2)*NUNS)/EE(1)
      NUTN=(EE(3)*NUNT)/EE(1)
      NUTS=(EE(3)*NUST)/EE(2)
      DIV=1.00-NUNS*NUSN-NUST*NUTS-NUNT*NUTN-NUSN*NUNT*NUTS
     1-NUNS*NUTN*NUST
      CNS(1,1)=EE(1)*(1.00-NUST*NUTS)/DIV
      CNS(1,2)=EE(2)*(NUNS+NUNT*NUTS)/DIV
      CNS(1,3)=EE(3)*(NUNT+NUNS*NUST)/DIV
      CNS(2,1)=CNS(1,2)
      CNS(2,2)=EE(2)*(1.00-NUNT*NUTN)/DIV
      CNS(2,3)=EE(3)*(NUST+NUSN*NUNT)/DIV
```

```
      CNS(3,1)=CNS(1,3)
      CNS(3,2)=CNS(2,3)
      CNS(3,3)=EE(3)*(1.00-NUNS*NUSN)/DIV
      CNS(4,4)=EE(7)
      CNS(5,5)=EE(8)
      CNS(6,6)=EE(9)
C     SET UP STRAIN TRANSFORM TO N-S-T SYSTEM
      SINA=SIN(ALPHA(M))
      COSA=COS(ALPHA(M))
      S2=SINA**2
      C2=COSA**2
      SC=SINA*COSA
      D(1,1)=C2
      D(1,3)=S2
      D(1,6)=-SC
      D(2,1)=S2
      D(2,3)=C2
      D(2,6)=SC
      D(3,2)=1.00
      D(4,1)=2.00*SC
      D(4,3)=-2.00*SC
      D(4,6)=C2-S2
      D(5,4)=SINA
      D(5,5)=COSA
      D(6,4)=COSA
      D(6,5)=-SINA
C     SET UP STRAIN TRANSFORMATION TO R-Z-T SYSTEM
      SINB=SIN(BETA(M))
      COSB=COS(BETA(M))
      S2=SINB**2
      C2=COSB**2
      SC=SINB*COSB
      C(1,1)=S2
      C(1,2)=C2
      C(1,4)=SC
      C(2,1)=C2
      C(2,2)=S2
      C(2,4)=-SC
      C(3,3)=1.00
      C(4,1)=-2.00*SC
      C(4,2)=2.00*SC
      C(4,4)=S2-C2
      C(5,5)=SINB
      C(5,6)=-COSB
      C(6,5)=COSB
      C(6,6)=SINB
C     CALCULATE CRZ MATRIX
      DO 120 I=1,6
      DO 120 J=1,6
      DUMMY(I,J)=0.00
```

```
      DO 120 K=1,6
  120 DUMMY(I,J)=DUMMY(I,J)+D(I,K)*C(K,J)
      DO 130 I=1,6
      DO 130 J=1,6
      DUMMY1(I,J)=0.00
      DO 130 K=1,6
  130 DUMMY1(I,J)=DUMMY1(I,J)+CNS(I,K)*DUMMY(K,J)
      DO 140 I=1,6
      DO 140 J=1,6
      DUMMY(I,J)=0.00
      DO 140 K=1,6
  140 DUMMY(I,J)=DUMMY(I,J)+D(K,I)*DUMMY1(K,J)
      DO 150 I=1,6
      DO 150 J=1,6
      CRZ(I,J)=0.00
      DO 150 K=1,6
  150 CRZ(I,J)=CRZ(I,J)+C(K,I)*DUMMY(K,J)
      WRITE(1) ((CRZ(I,J),J=1,6),I=1,6)
  151  CONTINUE
C
      DO 200 I=1,4
      MM= IX(N,I)
      XXX(I)= X(MM)
      XXX(I+4)= X(MM)
      YYY(I)= Y(MM)
  200 YYY(I+4)= Y(MM)
      XXX(9)= (XXX(1)+ XXX(2)+ XXX(3)+ XXX(4))/4.00
      YYY(9)= (YYY(1)+ YYY(2)+ YYY(3)+ YYY(4))/4.00
      XXX(10)= XXX(9)
      YYY(10)= YYY(9)
      DO 250 I=1,8
      CM(I)=0.00
  250 CONTINUE
      DO 900 IA=1,24
  900 S(IA)= 0.00
      CALL TRISTF(1,2)
      CALL TRISTF(2,3)
      CALL TRISTF(3,4)
      CALL TRISTF(4,1)
      RETURN
      END
      SUBROUTINE STIFF
      INTEGER CODE
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA,NCG
      COMMON/MATP/RO(12),E(9,12),EE(9),ETA(12)
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/NPDATA/X(200),Y(200),CODE(200),NPNUM(10,20)
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
      COMMON/RESULT/D(6,6),C(6,6),CNS(6,6)
```

```
      COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
      COMMON/BASIC2/BET,H,HN,HH(3),HT(2),TIME,NLAY,IFLAG,IT,NTIME
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
      COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
      COMMON/MASS/A(600),B(600),CM(8)
      COMMON/MASS1/XMINV(600),EEKM(13),F1(600),F2(600)
      IFLAG=1
      DO 50 I=1,IT
      GNM2(I)=GNM1(I)
      GNM1(I)=B(I)
   50 CONTINUE
      DO 100 I=1,IT
      B(I)= 0.00
  100 CONTINUE
      REWIND 1
      REWIND 2
      REWIND 3
      DO 340 N=1,NUMEL
      DO 340 M=1,NUMLA
      NLAY=M
      CALL QUAD
      DO 340 I=1,4
      II = 3*IX(N,I)+ 3*(M-1)*NUMNP
      J= II-2
      DO 340 K=J,II
      JJ=K-II+3*I
      B(K)= B(K)+S(JJ)
      KK= K+3*NUMNP
      B(KK)= B(KK)+S(JJ+12)
      IF(NTIME.GT.1)GO TO 340
      A(K)=A(K)+CM(I)
      A(KK)=A(KK)+CM(I+4)
  340 CONTINUE
      DO 400I=1,IT
 400  B(I)=-B(I)
      RETURN
      END
      SUBROUTINE STRESS
      INTEGER CODE
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA,NCG
      COMMON/NPR/NPRINT
      COMMON/MATP/RO(12),E(9,12),EE(9),ETA(12)
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/NPDATA/X(200),Y(200),CODE(200),NPNUM(10,20)
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
      COMMON/RESULT/D(6,6),C(6,6),CNS(6,6)
      COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
      COMMON/BASIC2/BET,H,HN,HH(3),HT(2),TIME,NLAY,IFLAG,IT,NTIME
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
```

```
      COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
      COMMON/MASS/A(600),B(600),CM(8)
      COMMON/DELT/XDEL(4,18)
      COMMON/DELTRI/DELTA(4,18)
      COMMON/PLYLD/SIGY(12,6),DEPS(6),EPS(12,50,6)
      COMMON/SIGZ/DSIGZ(12)
      DIMENSION DDEL(600)
C CALCULATE CHANGES IN DISPLACEMENTS
      DO 100 I=1,IT
      DDEL(I)=DEL(I)-DELN(I)
  100 CONTINUE
      DO 101 I=1,4
      DO 101 J=1,18
      XDEL(I,J)=0.00
      DELTA(I,J)=0.00
  101 CONTINUE
      IFLAG=2
      REWIND 1
      REWIND 2
      REWIND 3
C     WRITE(6,102)
  102 FORMAT("1   THESE ARE THE STRESSES CALCULATED FROM THE STRAINS AND
     1 PLASTIC YIELD WITHOUT MODIFICATION"/
     2 "     LAYER",2X,"ELEMENT",5X,"SIGMA X ",6X,"SIGMA Y ",6X,
     3 "SIGMA Z ",6X,"SIGMA XY",6X,"SIGMA YZ",6X,"SIGMA XZ")
      DO 200 N=1,NUMEL
      CALL ZSTRESS
      DO 200 M=1,NUMLA
      NLAY=M
      DO 150 I=1,4
      I1=I+1
      IF(I.EQ.4) I1=1
      MM=IX(N,I)
      MM1=IX(N,I1)
      II=3*MM+3*(M-1)*NUMNP
      II1=3*MM1+3*(M-1)*NUMNP
      J=II-2
      J1=II1-2
      DO 140 K=J,II
      IK=K-J+1
      XDEL(I,IK)=DDEL(K)
      DELTA(I,IK)=DELN(K)
      KK=K+3*NUMNP
      XDEL(I,IK+6)=DDEL(KK)
      DELTA(I,IK+6)=DELN(KK)
  140 CONTINUE
      DO 145 K=J1,II1
      IK=K-J1+1
      XDEL(I,IK+3)=DDEL(K)
      DELTA(I,IK+3)=DELN(K)
```

```
      KK=K+3*NUMNP
      XDEL(I,IK+9)=DDEL(KK)
      DELTA(I,IK+9)=DELN(KK)
  145 CONTINUE
  150 CONTINUE
C FIND DISPLACEMENT, BY AVERAGING, AT CENTER OF EACH ELEMENT
      XDEL(1,13)=(XDEL(1,1)+XDEL(2,1)+XDEL(3,1)+XDEL(4,1))/4.00
      XDEL(1,14)=(XDEL(1,2)+XDEL(2,2)+XDEL(3,2)+XDEL(4,2))/4.00
      XDEL(1,15)=(XDEL(1,3)+XDEL(2,3)+XDEL(3,3)+XDEL(4,3))/4.00
      XDEL(1,16)=(XDEL(1,7)+XDEL(2,7)+XDEL(3,7)+XDEL(4,7))/4.00
      XDEL(1,17)=(XDEL(1,8)+XDEL(2,8)+XDEL(3,8)+XDEL(4,8))/4.00
      XDEL(1,18)=(XDEL(1,9)+XDEL(2,9)+XDEL(3,9)+XDEL(4,9))/4.00
      DELTA(1,13)=(DELTA(1,1)+DELTA(2,1)+DELTA(3,1)+DELTA(4,1))
     1/4.00
      DELTA(1,14)=(DELTA(1,2)+DELTA(2,2)+DELTA(3,2)+DELTA(4,2))
     1/4.00
      DELTA(1,15)=(DELTA(1,3)+DELTA(2,3)+DELTA(3,3)+DELTA(4,3))
     1/4.00
      DELTA(1,16)=(DELTA(1,7)+DELTA(2,7)+DELTA(3,7)+DELTA(4,7))
     1/4.00
      DELTA(1,17)=(DELTA(1,8)+DELTA(2,8)+DELTA(3,8)+DELTA(4,8))
     1/4.00
      DELTA(1,18)=(DELTA(1,9)+DELTA(2,9)+DELTA(3,9)+DELTA(4,9))
     1/4.00
C CALCULATE AVERAGE STRAIN
      DO 175 I=2,4
      DO 175 J=13,18
      XDEL(I,J)=XDEL(1,J)
      DELTA(I,J)=DELTA(1,J)
  175 CONTINUE
  179 BSUM=0.00
      DO 180 I=1,6
      TSUM(I)=0.00
  180 CONTINUE
      CALL QUAD
      DO 190 I=1,6
      DEPS(I)=TSUM(I)/BSUM
  190 EPS(M,N,I)=EPS(M,N,I)+DEPS(I)
  999 CONTINUE
C CALCULATE STRESS INCREMENT
      DO 195 I=1,6
      DSIG(I)=0.00
      DO 195 J=1,6
  195 DSIG(I)= DSIG(I)+CRZ(I,J)*DEPS(J)
      CALL YIELD
  200 CONTINUE
      RETURN
      END
      SUBROUTINE TRISTF(II,JJ)
      INTEGER CODE
```

```
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA,NCG
      COMMON/MATP/RO(12),E(9,12),EE(9),ETA(12)
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/NPDATA/X(200),Y(200),CODE(200),NPNUM(10,20)
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
      COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
      COMMON/BASIC2/BET,H,HN,HH(3),HT(2),TIME,NLAY,IFLAG,IT,NTIME
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
      COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
      COMMON/MASS/A(600),B(600),CM(8)
      COMMON/DELT/XDEL(4,18)
      COMMON/DELTRI/DELTA(4,18)
      DIMENSION Q(6,18),C(18,18),XY(9,9),SAVE(18,3),BT(18,6),BTSIG(18),
     1CBAR(10,18),Z(10,10),          CBD(10),CC(10,4),BLT(18,4),
     2BADD(18,6)
      DIMENSION BLTSIG(18),BSIG(18)
      DIMENSION DUM(10,10)
      NLA=M
      NEL=N
      DO 50 I=1,6
   50 SIG(I)=SIGMA(M,N,I)
      THICK=TH(NLAY)
      ISUB= 9
      XX(1)= XXX(II)
      XX(2)= XXX(JJ)
      XX(3)= XXX(ISUB)
      YY(1)= YYY(II)
      YY(2)= YYY(JJ)
      YY(3)= YYY(ISUB)
       IF(NTIME.EQ.1) GO TO 40
      READ(2) ((BT(I,J),J=1,6),I=1,18)
        GO TO 801
   40  CONTINUE
  100 CALL INTER
  150 CONTINUE
      DO 300 IA=1,6
      DO 300 JA=1,18
  300 Q(IA,JA)=0.00
      Q(1,2)= XI(1)* THICK
      Q(1,11)= XI(1)*((THICK)**2)/2.00
      Q(3,18)= XI(3)* THICK
      Q(5,14)= XI(2)* THICK
      Q(2,6)= Q(1,2)
      Q(3,16)= Q(1,2)
      Q(3,17)= Q(5,14)
      Q(4,3)= Q(1,2)
      Q(4,5)= Q(1,2)
      Q(5,9)= Q(1,2)
      Q(5,13)= Q(1,2)
```

```
      Q(6,8)= Q(1,2)
      Q(6,10)= Q(1,2)
      Q(2,15)= Q(1,11)
      Q(4,12)= Q(1,11)
      Q(4,14)= Q(1,11)
      Q(5,18)= Q(1,11)
      Q(6,17)= Q(1,11)
      Q(5,15)= Q(3,18)
      Q(6,12)= Q(3,18)
      Q(6,11)= Q(5,14)
C
C THE FOLLOWING EXPRESSION IS ACTUALLY TWICE THE AREA OF THE TRIANGLE.
C
      AREA= XX(1)*YY(2)-XX(2)*YY(1)+XX(2)*YY(3)-XX(3)*YY(2)+XX(3)*YY(1)-
     1XX(1)*YY(3)
C
C     ZERO C MATRIX
C
      DO 400 I=1,18
      DO 400 J=1,18
  400 C(I,J)= 0.00
      FAC=1/AREA
      M=1
      N=1
      I= II
      J= JJ
      GO TO 250
   75 CONTINUE
      FAC=1/(AREA*THICK)
      M=10
      N=10
      I=II+4
      J=JJ+4
C
C   FILL C INVERSE BY PARTS
C
  250 C(M,N)= (XXX(J)* YYY(ISUB)- XXX(ISUB)* YYY(J))*FAC
      C(M,N+3)= (XXX(ISUB)* YYY(I)- XXX(I)* YYY(ISUB))*FAC
      C(M,N+6)= (XXX(I)* YYY(J)- XXX(J)* YYY(I))*FAC
      C(M+1,N)=(YYY(J)- YYY(ISUB))*FAC
      C(M+1,N+3)=(YYY(ISUB)- YYY(I))*FAC
      C(M+1,N+6)=(YYY(I)- YYY(J))*FAC
      C(M+2,N)=(XXX(ISUB)- XXX(J))*FAC
      C(M+2,N+3)=(XXX(I)- XXX(ISUB))*FAC
      C(M+2,N+6)=(XXX(J)- XXX(I))*FAC
      C(M+3,N+1)= C(M,N)
      C(M+3,N+4)= C(M,N+3)
      C(M+3,N+7)= C(M,N+6)
      C(M+4,N+1)= C(M+1,N)
      C(M+4,N+4)= C(M+1,N+3)
```

```
      C(M+4,N+7)= C(M+1,N+6)
      C(M+5,N+1)= C(M+2,N)
      C(M+5,N+4)= C(M+2,N+3)
      C(M+5,N+7)= C(M+2,N+6)
      C(M+6,N+2)= C(M,N)
      C(M+6,N+5)= C(M,N+3)
      C(M+6,N+8)= C(M,N+6)
      C(M+7,N+2)= C(M+1,N)
      C(M+7,N+5)= C(M+1,N+3)
      C(M+7,N+8)= C(M+1,N+6)
      C(M+8,N+2)= C(M+2,N)
      C(M+8,N+5)= C(M+2,N+3)
      C(M+8,N+8)= C(M+2,N+6)
      IF(ISUB.EQ.10) GO TO 200
      ISUB= 10
      GO TO 75
 200  M=NLA
      N=NEL
C
C   COMPUTING THE LOWER LEFT QUADRANT OF THE C MATRIX
C
      DO 600 I=1,9
      DO 600 J=1,9
      IJ=I+9
  600 C(IJ,J)=-C(I,J)/THICK
C
C  REARRANGE THE DISPLACEMENT MATRIX
C
      DO 730 IA=1,18
      DO 730 JA=1,3
      JB= JA+ 6
  730 SAVE(IA,JA)= C(IA,JB)
      DO 740 IA=1,18
      DO 740 JA=7,12
      JB=JA+3
  740 C(IA,JA)= C(IA,JB)
      DO 750 IA=1,18
      DO 750 JA=1,3
      JB=JA+12
  750 C(IA,JB)= SAVE(IA,JA)
C
C   CALCULATE THE BO TRANSPOSED MATRIX
C
      DO 800 JA=1,18
      DO 800 IA=1,6
      BT(JA,IA)=0.00
      DO 800 MA=1,18
  800 BT(JA,IA)=BT(JA,IA)+C(MA,JA)*Q(IA,MA)
       WRITE(2)((BT(I,J),J=1,6),I=1,18)
  801  CONTINUE
```

```
C
C   CALCULATE THE BOT * STRESS
C
      DO 810 IA=1,18
      BTSIG(IA)=0.00
      DO 810 NA=1,6
  810 BTSIG(IA)=BTSIG(IA)+BT(IA,NA)*SIG(NA)
C
C  ADD THE NONLINEAR TERMS
C
C
C   CALCULATE THE CBAR MATRIX
C
       IF(NTIME.EQ.1) GO TO 815
C LN IS THE NUMBER IN TIME STEPS BETWEEN CALCULATION FOR LARGE DEFORMATION
      LN=2
       IF((NTIME/LN )*LN.EQ. NTIME) GO TO 815
        READ(3) ((BLT(I,J),J=1,4),I=1,18)
       GO TO 861
  815 CONTINUE
      DO 820 IA=1,10
      DO 820 JA=1,18
      IB= IA+7
      IF(IA.EQ.9)IB=17
      IF(IA.EQ.10)IB=18
  820 CBAR(IA,JA)= C(IB,JA)
C
C   CALCULATE THE ZBAR MATRIX
C
      DO 830 IA=1,10
      DO 830 JA=1,10
  830 Z(IA,JA)= 0.00
      Z(1,1)= XI(1)*THICK
      Z(2,2)= Z(1,1)
      Z(3,3)= Z(1,1)
      Z(6,6)= Z(1,1)
      Z(3,4)= XI(2)*THICK
      Z(4,3)= Z(3,4)
      Z(6,7)= Z(3,4)
      Z(7,6)= Z(3,4)
      Z(9,9)= (THICK**3)/3*XI(1)
      Z(10,10)= Z(9,9)
      Z(1,9)= (THICK**2)/2*XI(1)
      Z(2,10)= Z(1,9)
      Z(9,1)= Z(1,9)
      Z(10,2)= Z(1,9)
      Z(3,5)= XI(3)* THICK
      Z(5,3)= Z(3,5)
      Z(6,8)= Z(3,5)
      Z(8,6)= Z(3,5)
```

```
      Z(4,4)= XI(5)* THICK
      Z(7,7)= Z(4,4)
      Z(5,5)= XI(6)* THICK
      Z(8,8)= Z(5,5)
      Z(4,5)= XI(4)* THICK
      Z(5,4)= Z(4,5)
      Z(7,8)= Z(4,5)
      Z(8,7)= Z(4,5)
C
C   CALCULATING THE CC MATRIX
C
      DO 840 IA=1,10
      CBD(IA)= 0.00
      DO 840 JA=1,18
  840 CBD(IA)= CBD(IA)+ CBAR(IA,JA)* DELTA(II,JA)
      DO 850 IA=1,10
      DO 850 JA=1,4
  850 CC(IA,JA)=0.00
      CC(1,1)= CBD(1)
      CC(2,4)= CBD(1)
      CC(1,4)= CBD(2)
      CC(2,2)= CBD(2)
      CC(3,3)= CBD(3)
      CC(4,3)= CBD(4)
      CC(5,3)= CBD(5)
      CC(6,3)= CBD(6)
      CC(7,3)= CBD(7)
      CC(8,3)= CBD(8)
      CC(9,1)= CBD(9)
      CC(10,4)= CBD(9)
      CC(9,4)= CBD(10)
      CC(10,2)= CBD(10)
      DO 855 I=1,10
      DO 855 J=1,4
      DUM(I,J)=0.00
      DO 855 K=1,10
      DUM(I,J)=DUM(I,J)+Z(I,K)*CC(K,J)
 855  CONTINUE
      DO 860 I=1,18
      DO 860 J=1,4
      BLT(I,J)=0.00
      DO 860 K=1,10
      BLT(I,J)=BLT(I,J)+CBAR(K,I)*DUM(K,J)
 860  CONTINUE
      WRITE(3) ((BLT(I,J),J=1,4),I=1,18)
  861  CONTINUE
      DO 870 IA=1,18
      BLTSIG(IA)= 0.00
      DO 870 JA=1,4
  870 BLTSIG(IA)= BLTSIG(IA)+ BLT(IA,JA)* SIG(JA)
```

```
      DO 880 IA=1,18
      BSIG(IA)= BTSIG(IA)+ BLTSIG(IA)
  880 CONTINUE
C  ADDING THE TRIANGULAR BSIG MATRICES TO FORM THE QUADRILATERAL MATRIX
      IF(IFLAG.NE.2) GO TO 1600
C EVALUATE STRAIN INCREMENT
      BSUM=BSUM+AREA/2.0*THICK
      DO 525 I=1,18
      DO 525 J=1,4
      BADD(I,J)= BT(I,J)+ BLT(I,J)
  525 CONTINUE
      DO 526 I=1,18
      DO 526 J=5,6
  526 BADD(I,J)= BT(I,J)
      DO 550 I=1,6
      DO 550 J=1,18
  550 TSUM(I)=TSUM(I)+BADD(J,I)*XDEL(II,J)
      GO TO 1000
 1600 CONTINUE
      K= 3*II-2
      L= 3*JJ-2
      DO 910 MA=1,2
      MJ= (MA-1)* 6
      NJ= (MA-1)* 12
      DO 920 IA=1,3
      JA= IA-1
      S(K+ NJ+ JA)= S(K+ NJ+ JA)+ BSIG(MJ+ IA)
  920 S(L+ NJ+ JA)= S(L+ NJ+ JA)+ BSIG(MJ+ IA+ 3)
      DO 910 IA=1,3
      DO 910 JA=1,4
      IJ= (JA-1)*3
  910 S(NJ+IJ+IA)= S(NJ+IJ+IA)+ BSIG(12+(MA-1)*3+IA)/4.0
      IF(NTIME.GT.1) GO TO 1000
C ASSEMBLE ELEMENT MASS MATRIX
      DEN=RO(MATRIL(NLAY))*THICK
      XMASS=AREA*DEN/8.00
      CM(II)=CM(II)+XMASS
      CM(JJ)=CM(JJ)+XMASS
      CM(II+4)=CM(II)
      CM(JJ+4)=CM(JJ)
 1000 CONTINUE
       RETURN
      END
      SUBROUTINE YIELD
      COMPLEX LAMBDA, DISC, DISQ
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
      COMMON/BASIC2/BET,H,HN,HH(3),HT(2),TIME,NLAY,IFLAG,IT,NTIME
      COMMON/PLYLD/SIGY(12,6),DEPS(6),EPS(12,50,6)
```

```
      COMMON/MATP/RO(12),E(9,12),EE(9),ETA(12)
      COMMON/SIGZ/DSIGZ(12)
      DIMENSION SIGT(6),SIGYB(3),TSIG(6),TSIGB(6)
      SIG3=SIGMA(M,N,3)+DSIGZ(M)
C FIND COMBINATION YIELD STRESSES (SIGYB)
      SIGYB(1)=1.0/SIGY(M,1)**2-1.0/SIGY(M,2)**2-1.0/SIGY(M,3)**2
      SIGYB(2)=1.0/SIGY(M,2)**2-1.0/SIGY(M,1)**2-1.0/SIGY(M,3)**2
      SIGYB(3)=1.0/SIGY(M,3)**2-1.0/SIGY(M,1)**2-1.0/SIGY(M,2)**2
C GET TEST STRESSES FROM ELASTIC ANALYSIS (SIGT)
      DO 15 I=1,6
   15 SIGT(I)=SIGMA(M,N,I)+DSIG(I)
C TEST FOR YIELDING BY HILL'S CRITERION
      YLD=0.0
      DO 20 I=1,6
   20 YLD=YLD+SIGT(I)**2/SIGY(M,I)**2
      YLD=YLD+SIGYB(1)*SIGT(2)*SIGT(3)+SIGYB(2)*SIGT(1)*SIGT(3)
     1 +SIGYB(3)*SIGT(1)*SIGT(2)
      IF(YLD.LT.1.0.OR.YLD.EQ.1.0) GO TO 390
C CORRECT STRESSES FOR PLASTICITY
C FIND TEST STRAIN INCREMENT VALUES (TSIG * LAMBDA = DEPS PLASTIC)
      TSIG(1)=SIGT(1)/SIGY(M,1)**2+(SIGYB(3)*SIGT(2)+SIGYB(2)*
     1 SIGT(3))/2.0
      TSIG(2)=SIGT(2)/SIGY(M,2)**2+(SIGYB(3)*SIGT(1)+SIGYB(1)*
     1 SIGT(3))/2.0
      TSIG(3)=SIGT(3)/SIGY(M,3)**2+(SIGYB(2)*SIGT(1)+SIGYB(1)*
     1 SIGT(2))/2.0
      DO 25 I=4,6
   25 TSIG(I)=SIGT(I)/(SIGY(M,I)**2)
      DO 35 I=1,6
      TSIGB(I)=0.0
      DO 35 J=1,6
   35 TSIGB(I)=TSIGB(I)+CRZ(I,J)*TSIG(J)
C FIND A,B,C OF THE QUADRADIC EQUATION TO CALCULATE LAMBDA
      AQ=0.0
      DO 45 I=1,6
   45 AQ=AQ+TSIGB(I)**2/SIGY(M,I)**2
      AQ=AQ+SIGYB(1)*TSIGB(2)*TSIGB(3)
     1 +SIGYB(2)*TSIGB(1)*TSIGB(3)
     2 +SIGYB(3)*TSIGB(1)*
     3 TSIGB(2)
      BQ=0.0
      DO 55 I=1,6
   55 BQ=BQ+TSIGB(I)*SIGT(I)/SIGY(M,I)**2
      BQ=BQ+(SIGYB(1)*(SIGT(2)*TSIGB(3)+SIGT(3)
     1 *TSIGB(2))+SIGYB(2)*(SIGT(1)*TSIGB(3)
     2 +SIGT(3)*TSIGB(1))+
     3 SIGYB(3)*(SIGT(1)*TSIGB(2)+SIGT(2)*TSIGB(1)
     4 ))/2.0
      CQ=YLD-1.0
      DISC=BQ**2-AQ*CQ
```

```
      DISQ=CSQRT(DISC)
      LAMBDA=CQ/(BQ+DISQ)
C CALCULATE ELASTIC-PLASTIC STRESSES
      DO 65 I=1,6
      SIGMA(M,N,I)=SIGMA(M,N,I)+DSIG(I)-TSIGB(I)*LAMBDA
   65 CONTINUE
      GO TO 400
  390 DO 395 I=1,6
  395 SIGMA(M,N,I)=SIGT(I)
      GO TO 600
  400 CONTINUE
C     WRITE(6,500)M,N,(SIGMA(M,N,I),I=1,6)
  500 FORMAT(2I8,2X,6E14.6)
      SIG3=(SIG3+SIGMA(M,N,3))/2.
      SIGMA(M,N,2)=SIGMA(M,N,2)-SIGMA(M,N,3)+SIG3
      SIGMA(M,N,1)=SIGMA(M,N,1)-SIGMA(M,N,3)+SIG3
      SIGMA(M,N,3)=SIG3
  600 CONTINUE
      RETURN
      END
      SUBROUTINE ZSTRESS
      INTEGER CODE
      COMMON/BASIC/VOL,NUMNP,NUMEL,NUMLA,NCG
      COMMON/MATP/RO(12),E(9,12),EE(9),ETA(12)
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/NPDATA/X(200),Y(200),CODE(200),NPNUM(10,20)
      COMMON/ELDATA/BETA(12),ALPHA(12),TH(12),IX(200,4),MATRIL(12)
      COMMON/TD/IMIN(100),IMAX(100),JMIN(25),JMAX(25),MAXI,MAXJ,NMTL,NBC
      COMMON/BASIC2/BET,H,HN,HH(3),HT(2),TIME,NLAY,IFLAG,IT,NTIME
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
      COMMON/DISP/DELN1(600),DELN(600),DEL(600),GNM1(600),GNM2(600)
      COMMON/MASS/A(600),B(600),CM(8)
      COMMON/DELT/XDEL(4,18)
      COMMON/FOR/FF(5),FC(5)
      COMMON/SIGZ/DSIGZ(12)
      COMMON/FIRST/FO(600),DER(600)
      DIMENSION AEI(12)
      IF(NTIME.NE.1)GO TO 15
      DO 16 I=1,6
   16 DSIGZ(I)=0.0
      GO TO 110
   15 CONTINUE
      AEIT=0.
      DO 20 I=1,4
      XXX(I)=X(IX(N,I))
      YYY(I)=Y(IX(N,I))
   20 CONTINUE
      XXX(5)=(XXX(1)+XXX(2)+XXX(3)+XXX(4))/4.0
      YYY(5)=(YYY(1)+YYY(2)+YYY(3)+YYY(4))/4.0
```

```
      FAV=0.
      CALL DISFOR
      DO 30 I=1,5
   30 FAV=FAV+FF(I)/5.
      DO 100 II=1,NUMLA
C COMPUTE AVERAGE ELEMENT ACCELERATION (AEA)
      I=NUMLA+1-II
      AEC1=0.
      DO 40 J=1,4
      DO 40 K=1,2
      INL=IX(N,J)*3+NUMNP*3*(I+1-K)
      IF(NTIME.EQ.1)GO TO 35
      AEC1=AEC1+(DEL(INL)-2.*DELN(INL)+DELN1(INL))/(8.*H**2)
      GO TO 40
   35 AEC1=AEC1+((DEL(INL)-DELN(INL))/H-DER(INL))/(8.*H)
   40 CONTINUE
      AEI(I)=AEC1
  100 AEIT=AEIT+AEC1/FLOAT(NUMLA)
C COMPUTE DIFFERENCE BETWEEN INDIVIDUAL ELEMENT ACCELERATION
C AND TOTAL PLATE ELEMENT ACCELERATION
      SART=0.
      DO 50 I=1,NUMLA
      MTR=MATRIL(I)
      RT=TH(I)*RO(MTR)
      SART=SART+RT*AEI(I)
   50 CONTINUE
      IF(SART.EQ.0.0)GO TO 52
      AK=FAV/SART
      GO TO 53
   52 AK=1.
   53 CONTINUE
C SOLVE FOR SIGMAZ
      SN=FAV
      DO 80 II=1,NUMLA
      I=NUMLA+1-II
      MTR=MATRIL(I)
      SNM=SN-AK*AEI(I)*RO(MTR)*TH(I)
      SIGZ=(SNM+SN)/2.0
      DSIGZ(I)=SIGZ-SIGMA(I,N,3)
   80 SN=SNM
  110 CONTINUE
      RETURN
      END
```

VISCOPLASTIC YIELD SUBROUTINE

```
      SUBROUTINE YIELD
      COMPLEX LAMBDA, DISC, DISQ
      COMMON/ARG/XXX(10),YYY(10),S(24),XX(3),YY(3),
     1CRZ(6,6),XI(10),SIG(12),N,M
      COMMON/SIGM/BSUM,TSUM(6),SIGMA(12,50,6),DSIG(6),F(600)
      COMMON/BASIC2/BET,H,HN,HH(3),HT(2),TIME,NLAY,IFLAG,IT,NTIME
      COMMON/PLYLD/SIGY(12,6),DEPS(6)
      COMMON/MATP/RO(12),E(9,12),EE(9),ETA(12)
      COMMON/SIGZ/DSIGZ(12)
      DIMENSION SIGT(6),SIGYB(3),TSIG(6),TSIGB(6)
      SIG3=SIGMA(M,N,3)+DSIGZ(M)
C FIND COMBINATION YIELD STRESSES (SIGYB)
      SIGYB(1)=1.0/SIGY(M,1)**2-1.0/SIGY(M,2)**2-1.0/SIGY(M,3)**2
      SIGYB(2)=1.0/SIGY(M,2)**2-1.0/SIGY(M,1)**2-1.0/SIGY(M,3)**2
      SIGYB(3)=1.0/SIGY(M,3)**2-1.0/SIGY(M,1)**2-1.0/SIGY(M,2)**2
C GET TEST STRESSES FROM ELASTIC ANALYSIS (SIGT)
      DO 15 I=1,6
   15 SIGT(I)=SIGMA(M,N,I)+DSIG(I)
C TEST FOR YIELDING BY HILL'S CRITERION
      YLD=0.0
      DO 20 I=1,6
   20 YLD=YLD+SIGT(I)**2/SIGY(M,I)**2
      YLD=YLD+SIGYB(1)*SIGT(2)*SIGT(3)+SIGYB(2)*SIGT(1)*SIGT(3)
     1 +SIGYB(3)*SIGT(1)*SIGT(2)
      IF(YLD.LT.1.0.OR.YLD.EQ.1.0) GO TO 390
C CORRECT STRESSES FOR PLASTICITY
C FIND TEST STRAIN INCREMENT VALUES (TSIG * LAMBDA = DEPS PLASTIC)
      TSIG(1)=SIGT(1)/SIGY(M,1)**2+(SIGYB(3)*SIGT(2)+SIGYB(2)*
     1 SIGT(3))/2.0
      TSIG(2)=SIGT(2)/SIGY(M,2)**2+(SIGYB(3)*SIGT(1)+SIGYB(1)*
     1 SIGT(3))/2.0
      TSIG(3)=SIGT(3)/SIGY(M,3)**2+(SIGYB(2)*SIGT(1)+SIGYB(1)*
     1 SIGT(2))/2.0
      DO 25 I=4,6
   25 TSIG(I)=SIGT(I)/(SIGY(M,I)**2)
      DO 35 I=1,6
      TSIGB(I)=0.0
      DO 35 J=1,6
   35 TSIGB(I)=TSIGB(I)+CRZ(I,J)*TSIG(J)
C FIND A,B,C OF THE QUADRADIC EQUATION TO CALCULATE LAMBDA
      AQ=0.0
      DO 45 I=1,6
   45 AQ=AQ+(TSIGB(I)+ETA(M)*SIGT(I)/H)**2/SIGY(M,I)**2
      AQ=AQ+SIGYB(1)*(TSIGB(2)+ETA(M)*SIGT(2)/H)*(TSIGB(3)+ETA(M)
     1 *SIGT(3)/H)+SIGYB(2)*(TSIGB(1)+ETA(M)*SIGT(1)/H)*(TSIGB(3)
     2 +ETA(M)*SIGT(3)/H)+SIGYB(3)*(TSIGB(1)+ETA(M)*SIGT(1)/H)*
     3 (TSIGB(2)+ETA(M)*SIGT(2)/H)
      BQ=0.0
      DO 55 I=1,6
   55 BQ=BQ+(TSIGB(I)+ETA(M)*SIGT(I)/H)*SIGT(I)/SIGY(M,I)**2
```

```
      BQ=BQ+(SIGYB(1)*(SIGT(2)*(TSIGB(3)+ETA(M)*SIGT(3)/H)+SIGT(3)
     1 *(TSIGB(2)+ETA(M)*SIGT(2)/H))+SIGYB(2)*(SIGT(1)*(TSIGB(3)
     2 +ETA(M)*SIGT(3)/H)+SIGT(3)*(TSIGB(1)+ETA(M)*SIGT(1)/H))+
     3 SIGYB(3)*(SIGT(1)*(TSIGB(2)+ETA(M)*SIGT(2)/H)+SIGT(2)*(TSIGB(1)
     4 +ETA(M)*SIGT(1)/H)))/2.0
      CQ=YLD-1.0
      DISC=BQ**2-AQ*CQ
      DISQ=CSQRT(DISC)
      LAMBDA=CQ/(BQ+DISQ)
C CALCULATE ELASTIC-PLASTIC STRESSES
      DO 65 I=1,6
   65 SIGMA(M,N,I)=(1.0+LAMBDA*ETA(M)/H)*(SIGT(I)-LAMBDA*(TSIGB(I)
     1 +ETA(M)*SIGT(I)/H))
      GO TO 400
  390 DO 395 I=1,6
  395 SIGMA(M,N,I)=SIGT(I)
      GO TO 600
  400 CONTINUE
      SIG3=(SIG3+SIGMA(M,N,3))/2.
      SIGMA(M,N,2)=SIGMA(M,N,2)-SIGMA(M,N,3)+SIG3
      SIGMA(M,N,1)=SIGMA(M,N,1)-SIGMA(M,N,3)+SIG3
      SIGMA(M,N,3)=SIG3
  600 CONTINUE
      RETURN
      END
```

EXAMPLE DATA INPUT

```
EXAMPLE---STEEL-ALUMINUM-STEEL 18X9 INCHES 05/29/79 DAT1C
    2    3   20    1
    1
        .25  2.50E-06  5.00E-06  1.00E-05  2.00E-05  6.00E-05
    4    6    4   35    3
  1  1      0.0     0.0  4  1     4.5     0.0                      1
  4  1      4.5     0.0  4  6     4.5     9.0                      1
  4  6      4.5     9.0  1  6     0.0     9.0                      1
  1  6      0.0     9.0  1  1     0.0     0.0                      1
    1    1         4
    2    3         2
    4    4         5
    5    5         1
    8    8         3
    9    9         1
   12   12         3
   13   13         1
   16   16         3
   17   17         1
   20   20         3
   21   21         6
   22   23         3
   24   24         7
   25   25         4
   26   28         2
   29   29         1
   33   33         1
   37   37         1
   41   41         1
   45   45         1
   49   49         4
   50   52         2
   53   53         1
   57   57         1
   61   61         1
   65   65         1
   69   69         1
   73   73         4
   74   76         2
   77   77         1
   81   81         1
   85   85         1
   89   89         1
   93   93         1
    1                  6.243388E+00
.3500000E+05
    2                  3.913917E-01
.5000000E+05
    1                  6.243388E+00
.3500000E+05
```

```
   1 .00073236
  3.0E+07   3.0E+07   3.0E+07        .25        .25        .25
  1.2E+07   1.2E+07   1.2E+07
   2 .00025251
  1.0E+07   1.0E+07   1.0E+07         .3         .3         .3
.3846E+07 .3846E+07 .3846E+07
 .25   .5   .5
GRID SIZE FOR THIS CASE IS 4 X 6 , 4/29/79, DAT1C
4.5       9.0
```

DISTRIBUTION LIST

No. of Copies	Organization
12	Commander Defense Technical Info Center ATTN: DDC-DDA Cameron Station Alexandria, VA 22314
1	Assistant Secretary of the Army (R&D) ATTN: Asst for Research Washington, DC 20310
1	HQDA (DAMA-ZA) Washington, DC 20310
1	HQDA (DAMA-AR) Washington, DC 20310
1	Commander US Army Materiel Development & Readiness Command ATTN: DRCDMD-ST 5001 Eisenhower Avenue Alexandria, VA 22333
2	Commander US Army Armament Research & Development Command ATTN: DRDAR-TSS (2 cys) Dover, NJ 07801
5	Commander US Army Armament Research & Development Command ATTN: DRDAR-LCA Mr. T. Stevens Mr. G. Randers-Pehrson Dr.N. Clark Mr. J. Hershkowitz Mr.J. Pearson Dover, NJ 07801
1	Commander US Army Armament Materiel Readiness Command ATTN: DRSAR-LEP-L, Tech Lib Rock Island, IL 61299
1	Director US Army ARRADCOM Benet Weapons Laboratory ATTN: DRDAR-LCB-TL Watervliet, NY 12189
1	Commander US Army Aviation Research & Development Command ATTN: DRSAV-E P.O. Box 209 St. Louis, MO 63166
1	Director US Army Air Mobility Research & Development Laboratory Ames Research Center Moffett Field, CA 94035
1	Commander US Army Communications Research & Development Command ATTN: DRDCO-PPA-SA Fort Monmouth, NJ 07703
1	Commander US Army Electronics Research & Development Command Technical Support Activity ATTN: DELSD-L Fort Monmouth, NJ 07703
1	Commander US Army Missile Command ATTN: DRSMI-R Redstone Arsenal, AL 35809
1	Commander US Army Missile Command ATTN: DRSMI-YDL Redstone Arsenal, AL 35809
1	Commander US Army Tank Automotive Research & Development Command ATTN: DRDTA-UL Warren, MI 48090

DISTRIBUTION LIST

No. of Copies	Organization
2	Commander US Army Materials and Mechanics Research Center ATTN: DRXMR-RF, J. Mescall Tech Lib Watertown, MA 02172
1	Commander US Army Research Office P.O. Box 12211 Research Triangle Park NC 27709
1	Director US Army TRADOC Systems Analysis Activity ATTN: ATAA-SL, Tech Lib White Sands Missile Range NM 88002
1	Commander Naval Research Laboratory Washington, DC 20375
2	Chief of Naval Research Department of the Navy ATTN: Code 427 Code 470 Washington, DC 20360
2	Commander Naval Air Systems Command ATTN: Code AIR-310 Code AIR-350 Washington, DC 20360
1	Commander Naval Ordnance Systems Command ATTN: Code ORD-0332 Washington, DC 20360
2	Commander Naval Surface Weapons Center ATTN: DG-50 DX-21, Lib Br Dahlgren, VA 22448

No. of Copies	Organization
1	Commander Naval Surface Weapons Center ATTN: Code 730, Lib Silver Spring, MD 20910
2	Commander Naval Weapons Center ATTN: Code 4057 Code 45, Tech Lib China Lake, CA 93555
1	USAF/AFRDDA Washington, DC 20311
1	AFSC/SDW Andrews AFB Washington, DC 20311
2	US Air Force Academy ATTN: Code FJS-41 (NC) Tech Lib Colorado Springs, CO 80840
1	AFATL/DLJW (J. Foster) Eglin AFB, FL 32542
1	AFWL/SUL (LT Tennant) Kirtland AFB, NM 87116
1	AFLC/MMWMC Wright-Patterson AFB, OH 45433
1	Director Lawrence Livermore Laboratory ATTN: Dr. J. Kury P.O. Box 808 Livermore, CA 94550
1	Director Lawrence Livermore Laboratory ATTN: Dr. M. Wilkins P.O. Box 808 Livermore, CA 94550

DISTRIBUTION LIST

No. of Copies	Organization
1	Director Lawrence Livermore Laboratory ATTN: Dr. E. Lee P.O. Box 808 Livermore, CA 94550
1	Director Lawrence Livermore Laboratory ATTN: Dr. H. Hornig P.O. Box 808 Livermore, CA 94550
1	Director Lawrence Livermore Laboratory ATTN: Dr.J. Knowles P.O. Box 808 Livermore, CA 94550
1	Director Lawrence Livermore Laboratory ATTN: Tech Lib P.O. Box 808 Livermore, CA 94550
1	Sandia Laboratories ATTN: Dr. W. Herrmann Albuquerque, NM 87115
1	Battelle Columbus Laboratories ATTN: Mr. Joseph E. Backofen 505 King Avenue Columbus, OH 43201
2	Dyna East Corporation ATTN: P.C.Chou J. Carleone 227 Hemlock Road Wynnewood, PA 19096
1	Honeywell, Inc. Government & Aeronautical Products Division ATTN: C.R. Hargreaves 600 Second Street, NE Hopkins, MN 55343

No. of Copies	Organization
1	Physics International ATTN: Dr.L. Behrmann 2700 Merced Street San Leandro, CA 94577
1	Shock Hydrodynamics ATTN: Dr.L. Zernow 4710-4716 Vineland Avenue North Hollywood, CA 91602
1	Systems, Science & Software ATTN: Dr.R. Sedgwick P.O. Box 1620 La Jolla, CA 92037
1	Drexel Institute of Technology Wave Propagation Research Center ATTN: Prof. P. Chou 32nd & Chestnut Streets Philadelphia, PA 19104
2	University of California Los Alamos Scientific Lab ATTN: Dr.J. Walsh Tech Lib P.O. Box 1663 Los Alamos, NM 87545
1	University of Denver Denver Research Institute ATTN: Mr.R.F. Recht 2390 S. University Boulevard Denver, CO 80210
3	University of Illinois AAE Department ATTN: Prof. A.R. Zak Urbana, Illinois 61801

Aberdeen Proving Ground

Dir, USAMSAA
ATTN: DRXSY-D; R.Bell, RAMD;
B.Oeheli; DRXSY-MP, H. Cohen
Cdr, USATECOM
ATTN: DRSTE-TO-F
Dir, Wpns Sys Concepts Team
Bldg E3516, EA
ATTN: DRDAR-ACW

USER EVALUATION OF REPORT

Please take a few minutes to answer the questions below; tear out this sheet and return it to Director, US Army Ballistic Research Laboratory, ARRADCOM, ATTN: DRDAR-TSB, Aberdeen Proving Ground, Maryland 21005. Your comments will provide us with information for improving future reports.

1. BRL Report Number ______________________________

2. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which report will be used.)

____

____

____

3. How, specifically, is the report being used? (Information source, design data or procedure, management procedure, source of ideas, etc.)______________________________

____

____

4. Has the information in this report led to any quantitative savings as far as man-hours/contract dollars saved, operating costs avoided, efficiencies achieved, etc.? If so, please elaborate.

____

____

5. General Comments (Indicate what you think should be changed to make this report and future reports of this type more responsive to your needs, more usable, improve readability, etc.)__________

____

____

____

6. If you would like to be contacted by the personnel who prepared this report to raise specific questions or discuss the topic, please fill in the following information.

Name:______________________________

Telephone Number:______________________________

Organization Address:______________________________

____

____